目　录

第一章　导　论

在过去的数十年中,数字信号处理(DSP)的领域无论在理论上还是技术上都有了非常重要的发展。它在工业中的成功是由于开发和利用了廉价的硬件和软件。在不同领域的新工艺和新应用现在都想利用 DSP 算法。这就导致对具有 DSP 基础的电气工程师的巨大需求。因此,很需要把 DSP 作为所有电气工程教学计划中的组成部分。

还在不久以前,DSP 的入门课程还主要是给研究生上的。它所附带的作业,如滤波器设计、谱估计、和有关的题目通常要在大型机或小型机上来做。然而过去十年中个人计算机及其软件的巨大进步,已经可能给本科教育加入一门 DSP 课程了。因为 DSP 的应用主要是在 DSP 处理器上[11]或软件中执行的一种算法,这就需要一定的编程量。利用交互式的软件,例如 MATLAB,就可以把重点放在学习新的困难的概念上而不在算法的编程上。还可以讨论许多有趣的实例和有用的习题。

根据这样的一种哲理,我们把这本书作为一本伴侣书(对于像[16,19]那样的传统的教科书而言)。其中 MATLAB 是所讨论的论题和概念的一个集成部分。我们把 MATLAB 选为编程工具主要因为它可广泛地在全国大学中的计算平台上获得。此外学生版的 MATLAB 已经流行了多年,它已成为最便宜的教学用软件。我们把 MATLAB 看作一个有若干工具的计算和编程工具箱(就像一个具有若干键的超级计算器),可以用它来探索和解决习题,因而能加强学习的进程。

本书写成入门的水平以便把本科学生引入 DSP 有趣的和实际的领域。我们强调这不是一本传统意义下的教科书而是一本伴侣书,因为它的重点是用 MATLAB 解题和动手的体验。我们假定学生已经熟悉了 MATLAB 并正在听 DSP 课程。本书提供了用数字技术处理现实世界信号(模拟信号)所需的基本分析工具。我们主要涉及离散信号和系统,并同时在时域和频域中分析。对滤波器和频谱分析器的结构的分析和设计是 DSP 中最重要的内容之一,在本书中作了详细的叙述。DSP 中的许多高级的论题(通常是在研究生课程中涉及)未在本书中讨论,但是我们希望在本书中获得的经验将帮助学生更方便和更深入地钻研高级的论题。

在本章中,我们将对 DSP 和 MATLAB 作一简单的综述。

数字信号处理的综述

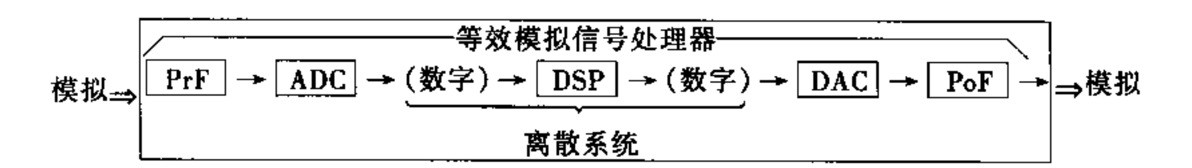
在现代的世界上,我们被不同形式的各种信号所包围。某些信号来自自然,而大部分的信号是人造的。有些信号是必须的(语言),有些是令人愉快的(音乐),很多则是在给定场合下不受欢迎或不需要的。因此从一大堆混合的杂乱的信息中提取或增强有用的信息是信号处理的最简单的形式。普遍地说,信号处理是设计来提取、增强、存储和传输有用信息的一种运算。有用和无用信息之间的区别往往是主观的,也有客观的。因此信号处理通常取决于应用。

信号是如何被处理的?

我们在实际中所遇到的信号大部分是模拟信号。它们的幅度随时间连续地变化。通常用由无源或有源器件组成的电网络对它们作处理。人所共知,这种方法称为模拟信号处理(ASP)。例如无线电和电视接收机。

模拟信号:$x_a(t)$ ⟹ 模拟信号处理器 ⟹ 模拟信号 $y_a(t)$

它们也可以用包括加法器、乘法器、和逻辑元件或用微处理器组成的数字硬件来处理。但是必须先把模拟信号变成适合于数字硬件来处理的形式。这样的信号形式称为数字信号。它在一定的时间间隔内只取有限数目的数值之一,因而它能用二进制数或位来表示。数字信号的处理称为 DSP,它可用框图表示:



图中各个方框的意义介绍如下:

PrF: 这是一个前置滤波器或反泄漏滤波器,它限制了模拟信号以防止泄漏。

ADC: 模-数转换器,它由模拟信号产生出一个二进制流。

数字信号处理器: 这是 DSP 的心脏,它可以代表一个通用计算机或专用处理器,或数字硬件等等。

DAC: 这是 ADC 的逆运算,称为数-模转换器,它由一个二进制流产生一个阶梯波形,作为形成模拟信号的第一步。

PoF: 这是一个后置滤波器,它把阶梯波形平滑成为预期的模拟信号。

从上面两种信号处理的方法中可以看到,DSP 方法比看来简单的 ASP 方法更加复杂,包括更多的元件。因此人们要问:为什么要用数字方法处理信号? 答案在于 DSP 带来的很多优点。

DSP 胜于 ASP 的优点

ASP 的主要缺点是在作复杂信号处理时只有有限的能力。这造成了处理的不灵活性和系统时间的复杂性。这也造成了产品的昂贵。另一方面,采用 DSP 方法,它可以把一个便宜的个人计算机变成一个强大的信号处理器。DSP 的一些重要的优点如下:

1. 用 DSP 方法的系统的开发可以用通用计算机上的软件来进行。因此 DSP 比较容易开发和测试,而软件是可移植的。

2. DSP 运算是单纯地基于加法和乘法,这导致特别稳定的处理性能——例如不受温度影响。

3. DSP 运算可方便地作实时的修改,通常只要改变程序,或对寄存器重新加载。

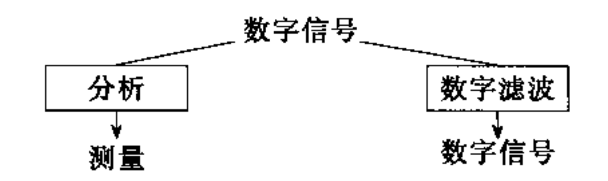
4. 由于超大规模集成电路(VLSI)降低了存储器、门、微处理器等等的价格,DSP 比较便

宜。

DSP 的主要缺点是运算速度低，特别是对很高频的信号。主要由于它的上述优点，DSP 现在已是许多技术和应用中的首选方法，诸如消费电子、通信、无绳电话和医学图象等。

DSP 的两个重要类别

大部分 DSP 运算可分为信号分析任务和信号滤波任务两类，图示如下：



信号分析　本任务涉及信号特性的测量。它通常是一个频域的运算。它的一些应用为：

* 谱(频率和/或相位)分析
* 语音分析
* 说话人识别
* 目标检测

信号滤波　本任务的特征是"信号入-信号出"的情况。实现这个任务的系统通常被称为滤波器。它通常(但不总是)作时域运算。它的一些用途为：

* 滤除不需要的背景噪声
* 去除干扰
* 频带分割
* 信号谱的成形

在某些应用中，例如语音合成，首先要对信号进行分析，研究它的特征。然后再用于数字滤波器中来产生合成的声音。

本书的前半部讨论 DSP 的信号分析部分。在第二章中，我们从离散信号和系统的描述开始。第三章对这些信号和系统在频域进行了分析。第四章介绍了频域描述的普遍方法——称为 z 变换。第五章讨论了计算付利叶变换的实际算法，包括离散付利叶变换和快速付利叶变换。

本书的后半部讨论 DSP 的信号滤波方面。在第六章中我们描述了数字滤波器的不同状态和结构。在第七章中给出了一种滤波器——有限长度脉冲响应(FIR)滤波器的设计技术和算法。而在第八章中给出了另一种称为无限长度脉冲响应(IIR)滤波器的设计方法。在这两章中，我们只讨论滤波器设计中简单而实用的技术，没有讨论更先进的技术。最后，在最末两章以设计任务的形式给出了一些实际应用题，它们可以用前八章所学的内容来完成。第九章介绍了自适应滤波，讨论了系统辨识、干扰抑制、自适应线路信号增强等等的简单设计任务。第十章给出了数字通信的一个简单的介绍，并给出有关 PCM、DPCM 和 LPC 等方面论题的设

计任务。

在所有各章中,中心主题是聪明地应用和适当地演示 MATLAB 工具。对已有的 DSP 中大多数的 MATLAB 函数都作了详细的说明,并在许多例子中演示了如何正确地使用它们。此外还开发了许多新的 MATLAB 函数,以便深入领会一些算法工作的内涵。我们相信,这种亲自动手的方法将能使学生去除对 DSP 的害怕心理并提供丰富的学习经验。

关于 MATLAB 的一些说明

MATLAB 是一种交互式的以矩阵为基础的系统,它用于科学和工程的计算与可视化。它的力量在于它只需用其他编程语言(如 FORTRAN 或 C)的几分之一的时间即可方便地解决复杂的数值计算问题。它的强有力也在于它那相对简单的编程功能。MATLAB 很容易扩展和创造新的命令和函数。

MATLAB 能在很多计算环境下运行:Sun/HP/VAXstation 工作站、80x86 PC、Apple Macintosh、VAX 和一些并行计算机。MATLAB 的基本程序已经由于近年来出现的大量的工具箱(即对某一专题开发的函数集)而得到进一步的加强。本书中的信息一般均适用于所有上述环境。本书的开发开始于在 DOS 下运行的 MATLAB 3.5 专业版。比较便宜包含较少的工具箱和函数的学生版也由 Prentice - Hall 出版社提供了。因此我们决定使本书中开发的 MATLAB 程序和其他内容能与学生版兼容。然而当前的主流版本已经是具有图形用户界面的 MATLAB 4.2 版,而且包含了增强了的新工具箱的具有图形用户界面(GUI)的新的学生版也已于 1995 年 2 月推出,本书也完全与这个版本兼容。我们尽了最大努力来弄清那些存在而可用的新函数。在学生版中新加了一个符号运算(Symbolic)工具箱,它是基于 MAPLE 的。因为数字信号处理主要用到数值计算,在本书中既未讨论也未用到符号运算工具箱。

MATLAB 的范围和能力远远超过了在本节中说的这些。再给出 MATLAB 的详细的信息和教材是没有意义的,因为在这个论题上已经有很多的优秀书籍和手册可用了。学生应该参阅 MATLAB 用户手册[2]和参考手册[1]。同样,学生应该尝试[3]中的教科书。上面所给的参考书中的信息,以及在线帮助的文件,通常对使用本书的学生已是足够了。

第二章　离散时间信号和系统

我们从离散时间的信号和系统开始研究。本章将介绍许多重要类型的信号和它们的运算。线性和时不变系统讨论得最多因为它们最容易分析和实现。对卷积和差分方程表示式给予了特别的注意,因为它们在数字信号处理和 MATLAB 中的重要性。本章的重点是用 MATLAB 表示和实现信号和系统。

离散时间信号

信号可以粗略地分为模拟信号和数字信号。模拟信号将用 $x(t)$表示,其中变量 t 可以表示任何物理量,但我们假定它代表以秒为单位的时间。离散信号用 $x(n)$表示。其中变量 n 为整数并代表时间的离散时刻。因此它也称为离散时间信号。他是一个数字的序列并可用以下符号之一来表述:

$$x(n)=\{x(n)\}=\{\ldots,x(-1),\underset{\uparrow}{x(0)},x(1),\ldots\}$$

其中,向上的箭头表示在 $n=0$ 处的取样。

在 MATLAB 中,我们可以用一个列向量来表示一个有限长度的序列。然而这样一个向量并没有包含采样位置的信息。因此,$x(n)$的正确表示方式应该包括分别表示 x 和 n 两个向量。例如序列 $x(n)=\{2,1,-1,\underset{\uparrow}{0},1,4,3,7\}$在 MATLAB 中表示为:

```
>>n=[-3,-2,-1,0,1,2,3,4]; x=[2,1,-1,0,1,4,3,7];
```

当不需要采样位置信息或这个信息是多余的时候(例如该序列从 $n=0$ 开始),我们可以只用 x 向量来表示。由于有限的内存,MATLAB 无法表示一个任意无限序列。

序列的类型

为了分析的需要,在数字信号处理中用了一些基本的序列。它们的定义和 MATLAB 表达式如下:

1.单位采样序列

$$\delta(n)=\begin{cases}1, & n=0\\0, & n\neq0\end{cases}=\left\{\ldots,0,0,\underset{\uparrow}{1},0,0,\ldots\right\}$$

在 MATLAB 中,函数 zeros(1,N)产生一个由 N 个零组成的列向量。它可用来实现有限

区间的 $\delta(n)$。然而,更高明的方法是利用逻辑关系式 n= =0 来实现 $\delta(n)$。例如,要实现

$$\delta(n-n_0)=\begin{cases}1, & n=n_0\\0, & n\neq n_0\end{cases}$$

在 $n_1\leqslant n_0\leqslant n_2$ 区间内的值,我们可用如下的 MATLAB 函数:

```
function [x,n] = impseq(n0,n1,n2)
% Generates x(n) = delta(n-n0); n1 <= n <= n2
% ------------------------------------------
% [x,n] = impseq(n0,n1,n2)
%
n = [n1:n2]; x = [(n-n0) == 0];
```

2.单位阶跃序列

$$u(n)=\begin{cases}1, n\geq 0\\0, n<0\end{cases}=\left\{\ldots,0,0,\underset{\uparrow}{1},1,1,\ldots\right\}$$

在 MATLAB 中函数 ones(1, N)产生由 N 个 1 组成的列向量。它可用来产生有限区间上的 $u(n)$。另一个高明的方法是使用逻辑关系式 $n>=0$。在 $n_1\leqslant n_0\leqslant n_2$ 区间内实现。

$$u(n-n_0)=\begin{cases}1, n\geq n_0\\0, n<n_0\end{cases}$$

可以用如下的 MATLAB 函数。

```
function [x,n] = stepseq(n0,n1,n2)
% Generates x(n) = u(n-n0); n1 <= n <= n2
% ------------------------------------------
% [x,n] = stepseq(n0,n1,n2)
%
n = [n1:n2]; x = [(n-n0) >= 0];
```

3.实数指数序列:

$$x(n)=a^n, \forall n; a\in R$$

在 MATLAB 中,要用数组运算符".^"来实现一个实指数序列。例如要实现序列 $x(n)=$

$(0.9)^n$, $0\leqslant n\leqslant 10$。应该用下列 MATLAB 程序：

```
>> n=[0:10]; x=(0.9).^n;
```

4.复数指数序列：

$$x(n)=e^{(\sigma+j\omega_0)n},\ \forall\ n$$

其中 σ 称为阻尼系数而 ω_0 是以弧度为单位的角频率。MATLAB 函数 exp 可用来产生指数序列。例如要产生 $x(n)=\exp[(2+j3)n]$,$0\leqslant n\leqslant 10$,可以用下列 MATLAB 程序：

```
>> n=[0:10]; x=exp((2+3j)*n);
```

5.正余弦序列：

$$x(n)=\cos(\omega_0 n+\theta),\ \forall\ n$$

其中 θ 为以弧度为单位的相角。MATLAB 函数 cos（或 sin)可用来产生正余弦序列。例如,要产生 $x(n)=3\cos(0.1\pi n+\pi/3)+2\sin(0.5\pi n)$, $0\leqslant n\leqslant 10$,应该用下列 MATLAB 程序。

```
>> n=[0:10]; x=3*cos(0.1*pi*n+pi/3) + 2*sin(0.5*pi*n);
```

6.随机序列：许多实际的序列是不能像上面那样用数学式来描述的。这些序列称为随机或 Stachastic 序列并用相应的概率密度函数或统计矩来表征。在 MATLAB 中,有两种(伪)随机序列可用。rand(1,N)产生其元素在[0,1]之间均匀分布而长度为 N 的随机序列。randn(1,N)则产生均值为 0,方差为 1,长度为 N 的高斯随机序列。其他的随机序列都可用这两种函数的变换得到。

7.周期序列：如果序列 $x(n)=x(n+N)$, $\forall\ n$ 则 $x(n)$称为周期的。满足上述关系的最小数 N 称为基本周期。我们用 $\tilde{x}(n)$来表示周期序列。要从一个周期$\{x(n),0\leqslant n\leqslant N-1\}$产生有 P 个周期的序列 $\tilde{x}(n)$,可以把它复制 P 次：

```
>> xtilde =[x,x,...,x];
```

但是高明的方法是利用 MATLAB 的强有力的下标能力。先产生一个包含 P 行 $x(n)$值的矩阵,然后用结构(:)来把它的 P 行串接起来成为一个长行,不过这种结构只能用于列向。所以我们往往还要用矩阵转置来把它扩展到行向。

```
>> xtilde=x' * ones(1,P);      % P列 x; x 是一个行向量
```

```
>> xtilde = xtilde(:);          % 长的列向量
>> xtilde = xtilde';            % 长的行向量
```

注意后两行可合成一条命令以缩短代码,这可从例 2.1 中看到。

对序列的运算

这里将简要地介绍序列的基本运算和它们的 MATLAB 等效程序。

1. 信号相加: 这是一种对应的样本与样本之间的相加。表为:

$$\{x_1(n)\} + \{x_2(n)\} = \{x_1(n) + x_2(n)\}$$

在 MATLAB 中它可用算术运算符"+"实现。然而 $x_1(n)$和 $x_2(n)$的长度必须相等。如果序列长度不等,或者长度虽然相等但采样的位置不同,就不能用运算符"+"了。我们必须首先给 $x_1(n)$和 $x_2(n)$ 以适当的参数使它们有同样的位置向量 n(因而也有同样的长度)。这需要非常注意 MATLAB 的下标运算。特别是逻辑与运算"&",关系运算如"<="和"==",以及要用 find 函数来使 $x1(n)$和 $x2(n)$具有相同的长度。下面称为 sigadd 的函数演示了这些运算。

```
function [y,n] = sigadd(x1,n1,x2,n2)
% 实现 y(n) = x1(n) + x2(n)
% ————————————————
% [y,n] = sigadd(x1,n1,x2,n2)
%    y = 在包括 n1 和 n2 的 n 上求和序列
% x1 = 长为 n1 的第一个序列
% x2 = 长为 n2 的第二个序列(n2 可与 n1 不等)
%
n = min(min(n1),min(n2)):max(max(n1),max(n2));        % y(n)的长度
y1 = zeros(1,length(n)); y2 = y1;                       %初始化
y1(find((n>=min(n1))&(n<=max(n1))==1)) = x1;           %具有 y(n)的长度的 x1
y2(find((n>=min(n2))&(n<=max(n2))==1)) = x2;           %具有 y(n)的长度的 x2
y = y1 + y2;                                            %序列相加
```

它的应用在例 2.2 中叙述。

2. 信号相乘: 这是对应采样值之间的相乘(即"点"乘),表为:

$$\{x_1(n)\} \cdot \{x_2(n)\} = \{x_1(n)x_2(n)\}$$

在 MATLAB 中,它由数组运算符".*"实现。但它受到"+"运算符同样的限制。因此必

须建立一个新的与 sigadd 相仿的 sigmult 函数：

```
function [y,n] = sigmult(x1,n1,x2,n2)
% 实现 y(n) = x1(n) * x2(n)
% ————————————————
% [y,n] = sigmult(x1,n1,x2,n2)
% y = 在包括 n1 和 n2 的 n 上求和序列
% x1 = 长为 n1 的第一个序列
% x2 = 长为 n2 的第二个序列(n2 可与 n1 不等)
%
n = min(min(n1),min(n2)):max(max(n1),max(n2));            % y(n)的长度
y1 = zeros(1,length(n)); y2 = y1;                          % 初始化
y1(find((n >= min(n1))&(n <= max(n1)) == 1)) = x1;  %具有 y(n)的长度的 x1
y2(find((n >= min(n2))&(n <= max(n2)) == 1)) = x2;  %具有 y(n)的长度的 x2
y = y1 .* y2;                                              % 序列相乘
```

其他用法也在例 2.2 中给出。

3.倍率:在此运算中,每个采样值乘以一个常数 α

$$\alpha\{x(n)\} = \{\alpha x(n)\}$$

在 MATLAB 中可用算术运算符“ * ”来实现倍率运算。

4. 移位:在这个运算中,$x(n)$的每一个样本都移动 k 个周期,移位后的序列 $y(n)$。

$$y(n) = \{x(n-k)\}$$

若令 $m=n-k$,因而 $n=m+k$,上述运算化成:

$$y(m+k) = \{x(m)\}$$

因为这一运算并不影响向量 x,但是向量 n 却因为每个元素都加了一个 k 而变化了。这些可在 sigshift 函数中看到:

```
function [y,n] = sigshift(x,m,n0)
% 实现 y(n) = x(n - n0)
% ————————————————
% [y,n] = sigshift(x,m,n0)
%
n = m + n0; y = x;
```

它的用法在例 2.2 中给出。

5.折叠:在这个运算中,$x(n)$的每个样本都对 $n=0$ 翻转,得到一个折叠后的序列 $y(n)$。

$$y(n)=\{x(-n)\}$$

在 MATLAB 中,这一运算对采样值由 fliplr(x)函数实现,而对采样位置则由 -fliplr(n)得到,这可从函数 sigfold 中看到。

```
function [y,n] = sigfold(x,n)
% 实现 y(n) = x(-n)
% ——————————————
% [y,n] = sigfold(x,n)
%
y = fliplr(x); n = -fliplr(n);
```

6.样本和:本运算和信号相加运算不同,它把 n_1 和 n_2 之间所有的样本 $x(n)$加起来:

$$\sum_{n=n_1}^{n_2} x(n)=x(n_1)+\cdots+x(n_2)$$

它由 sum($x(n1:n2)$)函数实现。

7.样本积:本运算和信号相乘运算不同,它把 n_1 和 n_2 之间所有的样本 $x(n)$乘起来:

$$\prod_{n_1}^{n_2} x(n)=x(n_1)\times\cdots\times x(n_2)$$

它由 prod($x(n1:n2)$)函数实现。

8.信号能量:序列 $x(n)$的能量由下式给出:

$$E_x=\sum_{-\infty}^{\infty} x(n)x^*(n)=\sum_{-\infty}^{\infty}|x(n)|^2$$

其中上标 * 表示共轭转置运算。有限长度序列的能量可用以下的 MATLAB 命令求得:

```
>> Ex = sum(x .* conj(x)); % 一种方法
>> Ex = sum(abs(x) .^ 2);  % 另一种方法
```

9.信号功率:基本周期为 N 的周期序列的平均功率由下式求得:

$$P_x = \frac{1}{N}\sum_{0}^{N-1}|x(n)|^2$$

□例 2.1:画出以下各序列在给定区间的波形图

a. $x(n) = 2\delta(n+2) - \delta(n-4)$, $-5 \leqslant n \leqslant 5$

b. $x(n) = n[u(n) - u(n-10)] + 10e^{-0.3(n-10)}[u(n-10) - u(n-20)]$, $0 \leqslant n \leqslant 20$

c. $x(n) = \cos(0.04\pi n) + 0.2\mathrm{w}(n))$, $0 \leqslant n \leqslant 50$ 其中 $\mathrm{w}(n)$为具有零均值及单位方差的高斯随机序列。

d. $\tilde{x}(n) = \{\ldots,5,4,3,2,1,\underset{\uparrow}{5},4,3,2,1,5,4,3,2,1\ldots\}$. $-10 \leqslant n \leqslant 9$

解: a. $x(n) = 2\delta(n+2) - \delta(n-4)$, $-5 \leqslant n \leqslant 5$

```
>>n=[-5:5];
>>x=2*impseq(-2,-5,5)-impseq(4,-5,5);
>>subplot(2,2,1);stem(n,x); title('例 2.1a 的序列')
>>xlabel('n');ylabel('x(n)');
```

图 2.1a 给出了本序列的波形。

b. $x(n) = n[u(n) - u(n-10)] + 10 * \exp(-0.3(n-10))[u(n-10) - u(n-20)]$, $0 \leqslant n \leqslant 20$

```
>>n=[0:20];
>>x1=n.*(stepseq(0,0,20)-stepseq(10,0,20));
>>x2=10*exp(-0.3*(n-10)).*(stepseq(10,0,20)-stepseq(20,0,20));
>>x=x1+x2;
>>subplot(2,2,2);stem(n,x);>>title('例 2.1b 的序列')
>>xlable('n');ylabel('x(n)');
```

图 2.1b 给出了本序列的波形。

c. $x(n) = \cos(0.04\pi n) + 0.2w(n)$, $0 \leqslant n \leqslant 50$.

```
>>n=[0:50];
>>x=cos(0.04*pi*n)+0.2*randn(size(n));
>>subplot(2,2,1);stem(n,x);title('例 2.1c 的序列')
>>xlabel('n');ylabel('x(n)');
```

图 2.1c 给出了本序列的波形。

d. $\tilde{x}(n) = \{\ldots,5,4,3,2,1,\underset{\uparrow}{5},4,3,2,1,\ldots\}$, $-10 \leqslant n \leqslant 9$

注意在给定的区间上序列 $x(n)$ 有四个周期。

```
>>n=[-10:9]; x=[5,4,3,2,1];
>>xtilde=x' * ones(1,4);
>>xtilde=(xtilde(:))';
>>subplot(2,2,4);stem(n,xtilde);title('例 2.1d 的序列')
>>xlabel('n');ylabel('xtilde(n)');
```

图 2.1d 给出了本序列的波形。

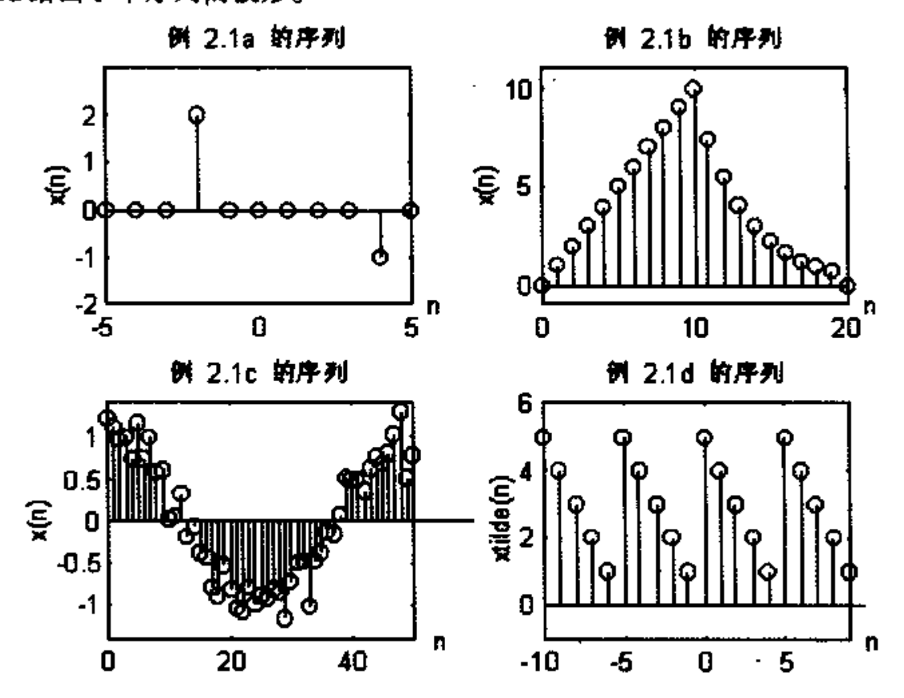


图 2.1　例 2.1 的序列曲线

□例 2.2　令 $x(n)=\{1,2,\underset{\uparrow}{3},4,5,6,7,6,5,4,3,2,1\}$，求出以下序列的波形：

a. $x_1(n)=2x(n-5)-3x(n+4)$

b. $x_2(n)=x(3-n)+x(n)x(n-2)$

解：序列 $x(n)$在 $-2\leqslant n\leqslant 10$ 区间内不等于零，因而

```
>>n=-2:10; x=[1:7,6:-1:1];
```

将产生 $x(n)$。

a. $x_1(n)=2x(n-5)-3x(n+4)$

其中第一部分由将 $x(n)$移位 5，第二部分将 $x(n)$移位 -4 求得。移位和相加可用 sigshift 和 sigadd 函数方便地完成。

```
>>[x11,n11]=sigshift(x,n,5); [x12,n12]=sigshift(x,n,-4);
>>[x1,n1]=sigadd(2*x11,n11,-3*x12,n12);
>>subplot(2,1,1); stem(n1,x1); title('例 2.2a 的序列')
```

```
>>xlabel('n');ylabel('x1(n)');
```

$x_1(n)$的波形见图 2.2a。

b. $x_2(n)=x(3-n)+x(n)x(n-2)$

其中第一项可以写成 $x(-(n-3))$。因而它由先把 $x(n)$折叠再把结果移位 3 而得。第二项为 $x(n)$和 $x(n-2)$的乘积,它们两者的长度相同但基底(或采样位置)不同。这一运算可方便地用函数 sigfold 和 sigmult 完成。

```
>>[x21,n21]=sigfold(x,n); [x21,n21]=sigshift(x21,n21,3);
>>[x22,n22]=sigshift(x,n,2); [x22,n22]=sigmult(x,n,x22,n22);
>>[x2,n2]=sigadd(x21,n21,x22,n22);
>>subplot(2,1,2); stem(n2,x2); title('例 2.2b 的序列')
>>xlabel('n');ylabel('x2(n)');
```

$x_2(n)$的波形见图 2.2b。

这个例子说明在本节中开发的四个 sig * 函数提供了一个处理序列的方便的方法。

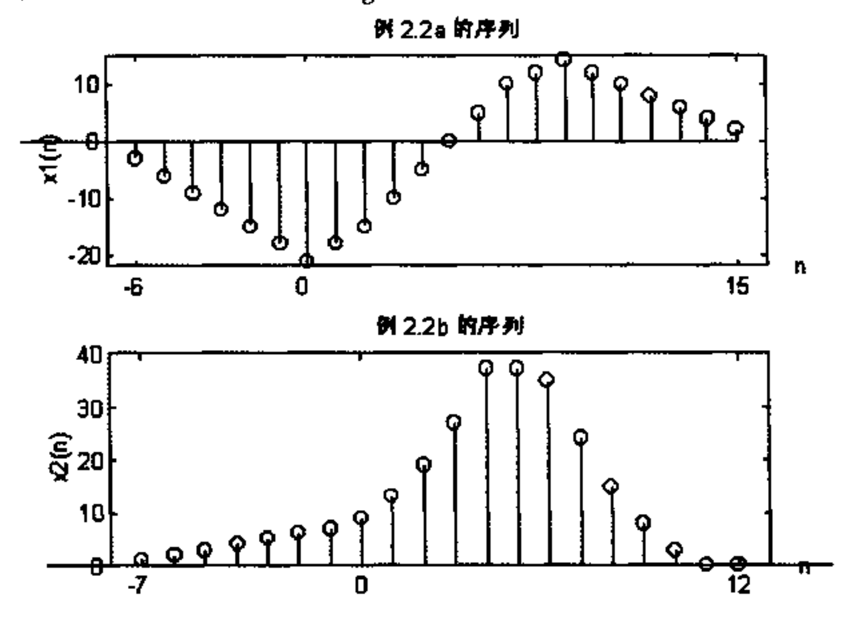


图 2.2 例 2.2 的序列

例 2.3 产生复数值信号:

$$x(n)=e^{(-0.1+j0.3)n}, \quad -10\leqslant n\leqslant 10$$

在四个子图中画出其幅度、相位、实部和虚部的波形。

解: 其 MATLAB 程序为:

```
>>n=[-10:1:10]; alpha=-0.1+0.3j;
```

```
> > x = exp(alpha * n);
> > subplot(2,2,1); stem(n,real(x));title('实部');xlable('n')
> > subplot(2,2,2); stem(n,imag(x));title('虚部');xlable('n')
> > subplot(2,2,3); stem(n,abs(x));title('振幅'); xlabel('n')
> >  subplot(2,2,4); stem(n,(180/pi) * angle(x)); title('相位'); xlabel('n')
```

此序列的波形见图 2.3。

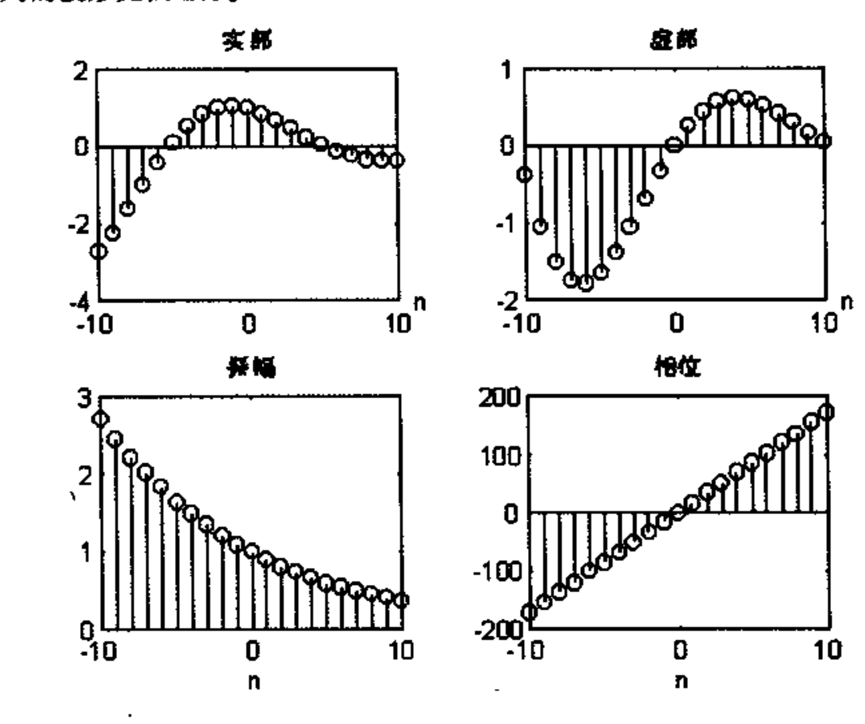


图 2.3 例 2.3 的复数序列分解

一些有用的结果

在离散时间信号理论中有不少重要的结果。我们将讨论在数字信号处理中有用的那些内容。

单位采样合成:任意序列 x(n)可用移位和倍率后的单位采样序列加权和求得:

$$x(n) = \sum_{k=-\infty}^{\infty} x(k)\delta(n-k) \tag{2.1}$$

我们将在下一节中用这个结果。

偶和奇合成:一个实数序列 $x_e(n)$被称为偶的(对称的)如果:

$$x_e(-n) = x_e(n)$$

同样,一个实数序列 $x_o(n)$被称为奇的(反对称的)如果:

$$x_o(-n) = -x_o(n)$$

这样任何任意实数序列 $x(n)$可用分解成它的偶和奇分量：

$$x(n) = x_e(n) + x_o(n) \tag{2.2}$$

其中偶部和奇部分别由下式给出：

$$x_e(n) = [x(n) + x(-n)]/2; \quad x_o(n) = [x(n) - x(-n)]/2 \tag{2.3}$$

我们将在研究付利叶变换的特性时用到这种分解。因此，开发一个简单的 MATLAB 函数来把一个给定信号分解为偶部和奇部是一个很好的练习。利用已有的 MATLAB 运算，我们可得到如下的 evenodd 函数。

```
function [xe, xo, m] = evenodd(x,n)
% 实信号分解为偶和奇两部分
% ——————————————
% [xe, xo, m] = evenodd(x,n)
%
if any(imag(x)~= 0)
    error('x 不是实序列')
end
m = -fliplr(n);
m1 = min([m,n]); m2 = max([m,n]); m = m1:m2;
nm = n(1) - m(1); n1 = 1:length(n);
x1 = zeros(1,length(m));
x1(n1 + nm) = x; x = x1;
xe = 0.5 * (x + fliplr(x));
xo = 0.5 * (x - fliplr(x));
```

本序列及其基底分别由 x 和 n 数组给出。程序首先检查给定的序列是否为实，并在 m 数组中确定偶和奇分量的基底。然后它执行(2.3)，特别注意 MATLAB 的下标运算。得出的分量分别存储在数组 xe 和 xo 中。

□**例 2.4** 设 $x(n) = u(n) - u(n-10)$，将它分解为偶和奇分量。

解：序列 $x(n)$称为矩形脉冲，它在 $0 <= n <= 9$ 区间不为零。我们将用 MATLAB 求得并画出其偶部和奇部。

```
>> n = [0:10]; x = stepseq(0,0,10) - stepseq(10,0,10);
>> [xe,xo,m] = evenodd(x,n);
>> figure(1);clf
```

```
>>subplot(2,2,1); stem(n,x); title('矩形脉冲')
>>xlabel('n'); ylabel('x(n)'); axis([-10,10,0,1.2])
>>subplot(2,2,2); stem(m,xe); title('偶部')
>>xlabel('n');ylabel('xe(n)'); axis([-10,10,0,1.2])
>>subplot(2,2,4); stem(m,xo); title('奇部')
>>xlabel('n'); ylabel('xe(n)'); axis([-10,10,-0.6,0.6])
```

图 2.4 清楚地显示了分解的结果。

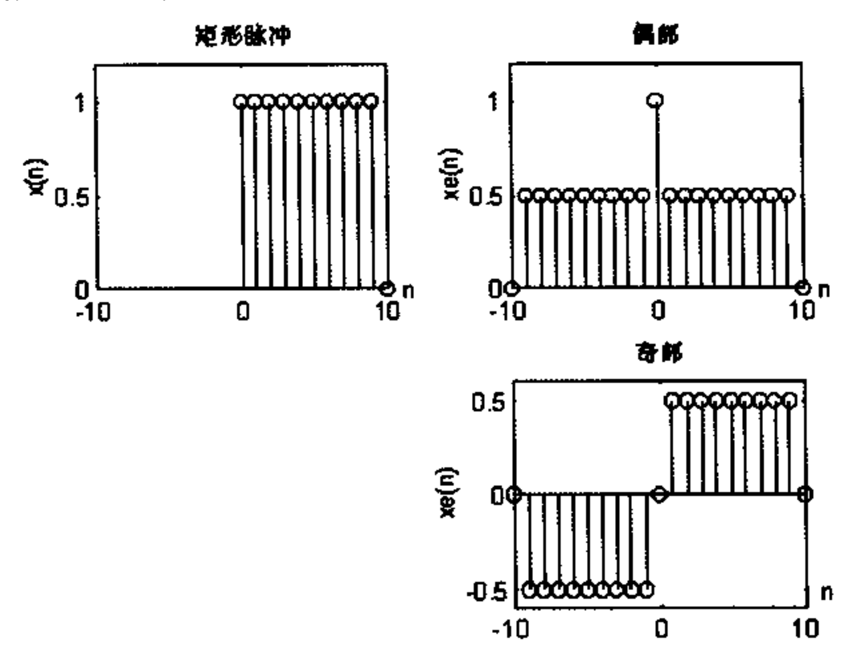


图 2.4 例 2.4 中的奇偶分解

在例 2.5 中给出了复数序列的一个类似的分解结果。

几何级数:形如$\{\alpha^n, n\geqslant 0\}$(其中 α 为任意常数)的单边指数序列称为几何级数。在数字信号处理的许多应用中要用到这个级数和的收敛性和表示式。该级数在$|\alpha|<1$时,其和收敛于:

$$\sum_{n=0}^{\infty}\alpha^n \longrightarrow \frac{1}{1-\alpha}, \qquad \text{对}|\alpha|<1 \tag{2.4}$$

我们也用得到该级数的有限项之和的表达式:

$$\sum_{n=0}^{N-1}\alpha^n = \frac{1-\alpha^N}{1-\alpha}, \forall \alpha \tag{2.5}$$

本书从头到尾要用到这两个结果。

序列的相关:数字信号处理应用中广泛用到相关运算。用它来度量两个序列相似的程度。给出两个长度相同,能量有限的序列 $x(n)$和 $y(n)$,则 $x(n)$和 $y(n)$的互相关是一个序列 $r_{xy}(l)$,其定义为:

$$r_{x,y}(l)=\sum_{n=-\infty}^{\infty}x(n)y(n-l) \tag{2.6}$$

下标 l 称为移位或滞后参数。当 $x(n)=y(n)$时,得到(2.6)式的特例,称为自相关。其定义为:

$$r_{xx}(l)=\sum_{n=-\infty}^{\infty}x(n)x(n-l) \tag{2.7}$$

它提供了序列位置不同对准情况下自相似程度的度量。用以计算自相关和互相关的 MATLAB 函数将在本章后面讨论。

离 散 系 统

数学上,一个离散时间系统(简称离散系统)可用算子 $T[.]$来描述。它取得一个序列 $x(n)$(称为激励)并将它变换为另一个序列 $y(n)$(称为响应)。即:

$$y(n)=T[x(n)]$$

在数字信号处理中,我们说系统将输入信号处理成输出信号。离散系统可粗略地分为线性和非线性系统。我们主要讨论线性系统。

线性系统

离散系统 $T[.]$是线性算子 $L[.]$,当且仅当 $L[.]$满足叠加原理,即

$$L[a_1x_1(n)+a_2x_2(n)]=a_1L[x_1(n)]+a_2L[x_2(n)],\ \forall a_1,a_2,x_1(n),x_2(n) \tag{2.8}$$

利用(2.1)和(2.8),线性系统在任意输入 $x(n)$下的输出 $y(n)$可表为:

$$y(n)=L[x(n)]=L\left[\sum_{n=-\infty}^{\infty}x(k)\delta(n-k)\right]=\sum_{n=-\infty}^{\infty}x(k)L[\delta(n-k)]$$

响应 $L[\delta(n-k)]$可以解释为由于一个 k 时刻的单位脉冲所引起的该线性系统在 n 时刻的响应。它被称作脉冲响应并表为 $h(n,k)$。因而输出可用叠加原理求得:

$$y(n)=\sum_{n=-\infty}^{\infty} x(k)h(n,k) \tag{2.9}$$

要算出(2.9)需要知道时变脉冲响应 $h(n,k)$,在实际上是不方便的。因此数字信号处理中广泛采用时不变(定常)系统。

线性时不变(LTI)系统:一个线性系统,其输入和输出对 $x(n),y(n)$不随时间移位 n 而变,称为线性时不变系统。对 LTI 系统,$L[.]$和移位算子是可逆的,表示如下:

$$x(n)\longrightarrow\boxed{L[\cdot]}\longrightarrow y(n)\longrightarrow\boxed{\text{移位 } k}\longrightarrow y(n-k)$$

$$x(n)\longrightarrow\boxed{\text{移位 } k}\longrightarrow x(n-k)\longrightarrow\boxed{L[\cdot]}\longrightarrow y(n-k)$$

我们用算子 LTI[.]表示 LTI 系统。令 $x(n)$和 $y(n)$为一个 LTI 系统的输入输出对。则时变函数 $h(n,k)$成为一个时不变函数 $h(n-k)$,而(2.9)式得出的输出成为:

$$y(n)=\mathrm{LTI}[x(n)]=\sum_{k=-\infty}^{\infty} x(k)h(n-k) \tag{2.10}$$

LTI 系统的脉冲响应为 $h(n)$.(2.10)式的数学运算称为线性卷积和并表为:

$$y(n)\triangleq x(n)*h(n) \tag{2.11}$$

由此,一个 LTI 系统可以完全用脉冲响应 $h(n)$在时域中表征,如下所示:

$$x(n)\longrightarrow\boxed{h(n)}\longrightarrow y(n)=x(n)*h(n)$$

我们将在题 2.12 中讨论卷积的若干特性。

稳定性:在线性系统中,这是一个很重要的概念。考虑稳定性的主要理由是避免构成有害的系统或者避免系统工作时发生损害或饱和。如果系统在所有的有界输入下都产生有界的输出:

$$|x(n)|<\infty\Rightarrow|y(n)|<\infty,\ \forall x,y$$

则称它是有界输入-有界输出(BIBO)稳定的 。一个 LTI 系统是 BIBO 稳定的,当且仅当其脉冲响应是绝对可加的。

$$\text{BIBO 稳定性}\Leftrightarrow\sum_{-\infty}^{\infty}|h(n)|<\infty \tag{2.12}$$

因果性:要想保证系统可以造出,必须有这个重要概念。如果系统在下标 n_0 处的值仅仅依赖于 n_0 点及它之前的输入,也就是它的输出不依赖于未来的输入,则系统被称为因果性的。一个 LTI 系统是因果的,当且仅当其脉冲响应:

$$h(n)=0, \quad n<0 \tag{2.13}$$

这一序列称为因果序列。在信号处理中,除非有专门的说明,总是假定系统的因果的。

卷　积

我们引入了卷积运算来描述一个 LTI 系统的响应。在数字信号处理中它是一个重要的运算,在本书中会看到它有很多别的用处。可以有很多方法来求卷积。如果序列是数学函数(有限或无限长度),则可用分析的方法求(2.11),得到对所有的 n 都适用的 $y(n)$ 函数形式。

□**例 2.5**: 将例 2.4 中的矩形脉冲 $x(n)=u(n)-u(n-10)$ 作为对一个脉冲响应为

$$h(n)=(0,9)^n u(n)$$

的 LTI 系统的输入,求输出 $y(n)$。

解:输入 $x(n)$ 及脉冲响应 $h(n)$ 示于图 2.5 中。由(2.11)

$$y(n)=\sum_{k=0}^{9}(1)(0.9)^{(n-k)}u(n-k)=(0.9)^n\sum_{k=0}^{9}(0.9)^{-k}u(n-k) \tag{2.14}$$

(2.14)中的总和差不多是一个几何级数和,除了 $u(n-k)$ 项依赖于不同的 n 和 k 而取值。$u(n-k)$ 可能在三种不同的情况下取值:

情况ⅰ $n<0$: 此时 $u(n-k)=0,\ 0\leq k\leq 9$。因而从(2.14)得

$$y(n)=0 \tag{2.15}$$

在此情况下 $x(n)$ 和 $h(n)$ 的非零值不互相覆盖。

情况ⅱ $0\leq n<9$: 则 $u(n-k)=1, \quad 0\leq k\leq n$。因而由(2.14)

$$\begin{aligned} y(n)&=(0.9)^n\sum_{k=0}^{n}(0.9)^{-k}=(0.9)^n\sum_{k=0}^{n}[(0.9)^{-1}]^k \\ &=(0.9)^n\frac{1-(0.9)^{-(n+1)}}{1-(0.9)^{-1}}=10[1-(0.9)^{n+1}],0\leq n<9 \end{aligned} \tag{2.16}$$

在此情况下,脉冲响应 $h(n)$ 与输入 $x(n)$ 部分覆盖。

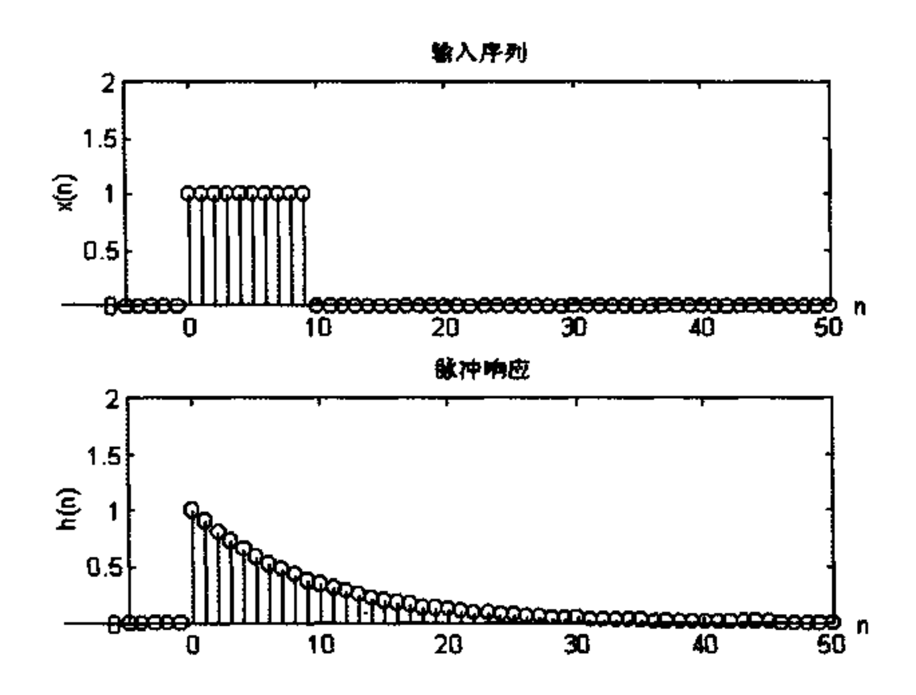


图 2.5 例 2.5 中的输入序列和脉冲响应序列

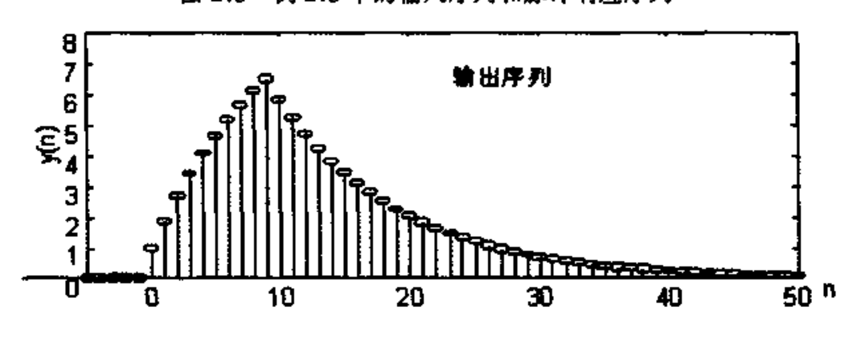


图 2.6 例 2.5 的输出序列

情况 iii $n\geq 9$：故 $u(n-k)=1,\ 0\leq k\leq 9$，并由(2.14)

$$y(n)=(0.9)^{n}\sum_{k=0}^{9}(0.9)^{-k} \tag{2.17}$$

$$=(0.9)^{n}\frac{1-(0.9)^{-10}}{1-(0.9)^{-1}}=10(0.9)^{n-9}[1-(0.9)^{10}],n\geq 9$$

在此情况下，$h(n)$完全与 $x(n)$覆盖。

由(2.15)、(2.16)和(2.17)给出了完整的响应式。它表示于在图 2.6 中，显示了输入脉冲经过系统后发生了畸变。

上面的例子也可以用一种被称为图解卷积的方法来求得，这时(2.11)要用一种图解说明。在这个方法中 $h(n-k)$要解释为折叠-移位了的 $h(k)$。输出 $y(n)$是靠在 $x(k)$和 $h(n-k)$覆盖区内样本和求得的。我们用一个例子来说明。

□**例 2.6:**给出以下两个序列:

$$x(n)=[3,11,7,\underset{\uparrow}{0},-1,4,2],-3\leq n\leq 3;\quad h(n)=[2,\underset{\uparrow}{3},0,-5,2,1],-1\leq n\leq 4$$

试求其卷积 $y(n)=x(n)*h(n)$

解:在图 2.7 中给出了四个图。上左图示出原始序列 $x(k)$和 $h(k)$。上右图示出$x(k)$和把 $h(k)$折叠后的 $h(-k)$。下左图给出 $x(k)$和把 $h(k)$折叠并移位 -1 后的$h(-1-k)$。于是:

$$\sum_k x(k)h(-1-k)=3\times(-5)+11\times 0+7\times 3+0\times 2=6=y(-1)$$

下右图给出 $x(k)$ 和把 $h(k)$折叠并移位 2 后的 $h(2-k)$。因而得出:

$$\sum_k x(k)h(2-k)=11\times 1+7\times 2+0\times(-5)+(-1)\times 0+4\times 3+2\times 2=41=y(2)$$

这样我们得出了 $y(n)$的两个值。$y(n)$的其他值也可用图解积分法类似地求得。注意 $y(n)$的起始点(第一个非零样本点)为 $n=-3+(-1)=-4$,而终点(末一个非零点)为 $n=3+4=7$。完整的输出为:

$$y(n)=\{6,31,47,6,-5\underset{\uparrow}{1},-5,41,18,-22,-3,8,2\}$$

希望学生来校验上述结果。注意产生的序列 $y(n)$比 $x(n)$和 $h(n)$两个序列都要长。

MATLAB 实现

如果任意序列是无限长度的,就不能用 MATLAB 来直接计算卷积。MATLAB 确实提供了一个内部函数 conv 来计算两个有限长度序列的卷积。conv 函数假定两个序列都从 $n=0$ 开始。调用如下:

```
>>y=conv(x,h)
```

例如,要求例 2.5 中的卷积,我们可用:

```
>> x=[3,11,7,0,-1,4,2];
>> h=[2,3,0,-5,2,1];
>> y=conv(x,h)
y =
   6  31  47  6  -51  -5  41  18  -22  -3  8  2
```

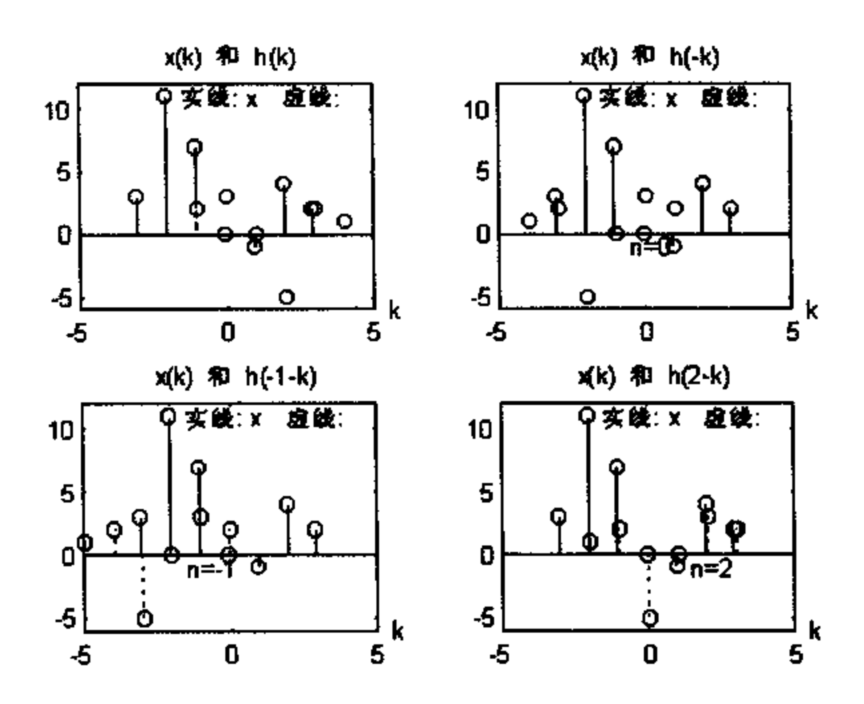


图 2.7 例 2.6 的图解卷积

以得到 $y(n)$的正确值。然而，如果序列具有任意基底，函数 conv 既不给出也不接受任何时间信息。需要知道什么才能得到 $y(n)$的起点和终点呢？给出有限序列 $x(n)$和 $h(n)$，不难求出这两个点.令

$$\{x(n); n_{xb} \leqslant n \leqslant n_{xe}\} \quad 及 \quad \{h(n);\ n_{hb} \leqslant h \leqslant n_{he}\}$$

为有限序列。参考例 2.6，可得到 $y(n)$的起点和终点分别为：

$$n_{yb} = n_{xb} + n_{hb} \quad 及 \quad n_{ye} = n_{xe} + n_{he}$$

将函数 conv 稍加扩展为函数 conv_m，它可以对任意基底的序列求卷积。程序如下：

```
function [y,ny] = conv_m(x,nx,h,nh)
% 信号处理的改进卷积程序
% ---------------------
% [y,ny] = conv_m(x,nx,h,nh)
% [y,ny] = 卷积结果
% [x,nx] = 第一个信号
% [h,nh] = 第二个信号
%
nyb = nx(1) + nh(1); nye = nx(length(x)) + nh(length(h));
ny = [nyb:nye];
```

```
y = conv(x,h);
```

□ 例 2.7 用 conv _ m 求例 2.6 中的卷积。

解:MATLAB 程序

```
>>x = [3, 11, 7, 0, -1, 4, 2]; nx = [-3: 3];
>>h = [2, 3, 0, -5, 2, 1]; ny = [-1: 4];
>>[y,ny] = conv _ m(x,nx,h,nh)
y =
  6  31  47  6  -51  -5  41  18  -22  -3  8  2
ny =
  -4  -3  -2  -1  0   1  2  3  4  5  6  7
```

因而和例 2.6 一样,

$$y(n) = \{6,31,47,6,-\underset{\uparrow}{51},-5,41,18,-22,-3,8,2\}$$

在 MATLAB 中还有一种方法可用来求卷积。这个方法利用了矩阵-向量相乘的算法,我们将在习题 2.13 中探讨。

再谈序列相关:

如果我们把卷积运算和例 2.6 中两序列的相关运算作一比较,可以看到非常相似。互相关可以写成如下形式:

$$r_{yx}(l) = y(l) * x(-l)$$

自相关可以写成如下形式:

$$r_{xx}(l) = x(l) * x(-l)$$

由此如果序列具有有限长度的话,这些相关可以用 conv 函数来求。

□例 2.8　在本例中,我们将演示互相关序列的应用。令

$$x(n) = [3,11,7,\underset{\uparrow}{0},-1,4,2]$$

为一原型序列,令 $y(n)$为 $x(n)$加入噪声干扰并移位后的序列:

$$y(n) = x(n-2) + w(n)$$

其中 $w(n)$为具有零均值和单位方差的高斯序列。计算 $x(n)$和 $y(n)$之间的互相关。

解：由 $y(n)$的结构可以看出它“相似于”$x(n-2)$，因而它们之间的相关应该在 $l=2$ 处显示出最强。为了用 MATLAB 测试出这一点，用两个不同的噪声来计算其互相关。

```
% 噪声序列 1
>>x=[3, 11, 7, 0, -1, 4, 2]; nx=[-3:3];          % 给定信号 x(n)
>>[y,ny]=sigshift(x,nx,2);                        % 得到 x(n-2)
>>w=randn(1,length(y)); nw=ny;                    % 产生 w(n)
>>[y,ny]=sigadd(y,ny,w,nw);                       % 得到 y(n)=x(n-2) + w(n)
>>[x,nx]=sigfold(x,nx);                           % 得到 x(-n)
>>[rxy,nrxy]=conv_m(y,ny,x,nx);                   % 互相关
>>subplot(1,1,1),subplot(2,1,1);stem(nrxy,rxy)
>>axis([-5,10,-50,250]);xlabel('迟延量 l')
>>ylabel('rxy');title('噪声序列 1 的互相关')
%
% 噪声序列 2
>>x=[3, 11, 7, 0, -1, 4, 2]; nx=[-3:3];          % 给定 x(n)
>>[y,ny]=sigshift(x,nx,2);                        % 得到 x(n-2)
>>w=randn(1,length(y)); nw=ny;                    % 产生 w(n)
>>[y,ny]=sigadd(y,ny,w,nw);                       % 得到 y(n)=x(n-2) + w(n)
>>[x,nx]=sigfold(x,nx);                           % 得到 x(-n)
>>[rxy,nrxy]=conv_m(y,ny,x,nx);                   % 互相关
>>subplot(2,1,2);stem(nrxy,rxy)
>>axis([-5,10,-50,250]);xlabel('迟延量 l')
>>ylabel('rxy');title('噪声序列 2 的互相关')
```

从图 2.8 我们看到互相关的峰值确实在 $l=2$ 处。这说明 $y(n)$和 $x(n)$移位 2 后相似。这个方法可以在雷达信号处理中用以对目标进行识别和局部化。

要注意 MATLAB 的信号处理工具箱中也提供了一个称为 xcorr 的函数来进行序列的相关计算。它的最简单的形式为：

```
>> xcorr(x,y)
```

用以计算向量 x 和 y 之间的互相关。而

```
>> xcorr(x)
```

则用以计算向量 x 的自相关。在 MATLAB 学生版中没有这个函数。它产生的结果和正确使用 conv_m 所得结果相同。不过，xcorr 函数不能像 conv_m 函数那样提供时间(或迟延)信息，必须辅以其他手段。因此我们还是强调应用 conv_m 函数。

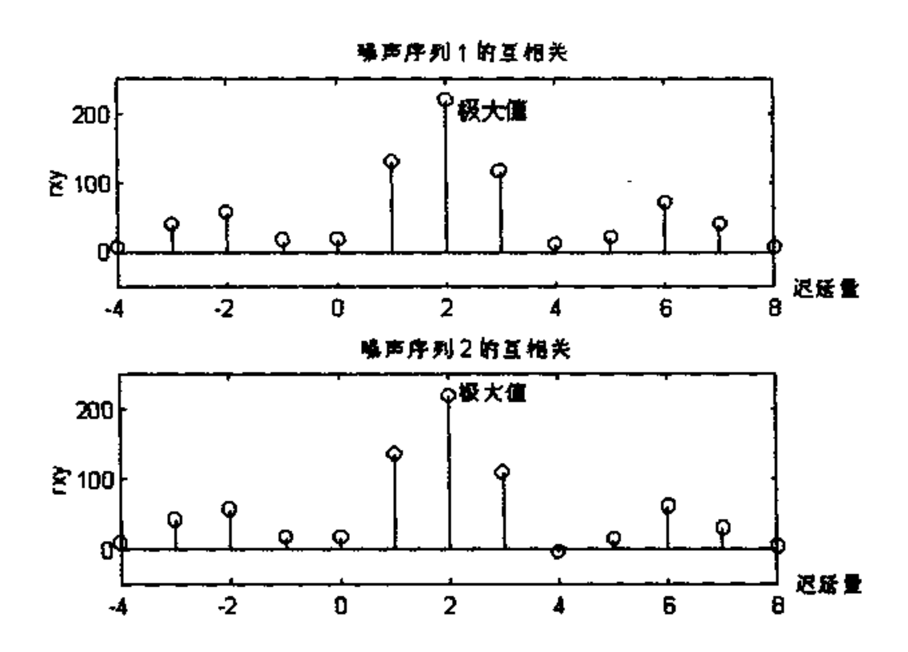


图 2.8　用两个不同的噪声实现求出的互相关序列

差 分 方 程

LTI 离散系统也能用下列形式的线性长系数差分方程来描述：

$$\sum_{k=0}^{N} a_k y(n-k) = \sum_{m=0}^{M} b_m x(n-m), \forall n \tag{2.18}$$

如果 $a_N \neq 0$,则此差分方程是 N 阶的。本方程描述了根据输入和以前的输出来计算当前输出的一个递推方法。实际上这个方程是按时间向前推算的,从 $n=-\infty$ 到 $n=\infty$。因此本方程的另一个形式为：

$$y(n) = \sum_{m=0}^{M} b_m x(n-m) - \sum_{k=1}^{N} a_k y(n-k) \tag{2.19}$$

这个方程的解可以写成

$$y(n) = y_H(n) + y_P(n)$$

其通解部分 $y_H(n)$为：

$$y_H(n) = \sum_{k=1}^{N} c_k z_k^n$$

其中 $z_k, k = 1,2,\ldots,N$ 是特征方程

$$\sum_{0}^{N} a_k z^k = 0$$

的 N 个根。特征方程在确定系统稳定性时是很重要的。如果这些根满足下述条件：

$$|z_k| < 1, \qquad k = 1,2,\ldots,N \tag{2.20}$$

则由(2.19)描述的因果系统是稳定的。特解部分 $y_P(n)$由(2.18)式的右端确定。在第四章中将讨论用 z 变换解差分方程的方法。

MATLAB 实现

有一个称为 filter 的子程序专用来在给定输入和差分方程系数时求差分方程的数值解。子程序调用的最简单形式为：

```
y = filter(b,a,x)
```

其中

```
b=[b0, b1, ..., bM]; a=[a0, a1, ..., aN];
```

是由方程(2.18)中给出的系数数组，而 x 则是输入序列数组。输出 y 和输入 x 的长度一样，必须保证系数 $a0$ 不为零。我们用下列例子来说明其用法。

□ **例 2.9**　给出如下差分方程：

$$y(n) - y(n-1) + 0.9y(n-2) = x(n); \quad \forall n$$

a. 计算并画出脉冲响应 h(n), n = -20,...,100.
b. 计算并画出阶跃响应 s(n), n = -20,...,100.
c. 由此 h(n)规定的系统是否稳定？

解：由给定的差分方程可知，其系数数组为：

a. MATLAB 程序

```
>> b = [1];a = [1, -1,0.9];
```

```
>>x = impseq(0, -20,120);n = [-20:120];
>>h = filter(b,a,x);
>>subplot(2,1,1);stem(n,h)
>>axis([-20,120,-1.1,1.1])
>>title('脉冲响应');xlabel('n');ylabel('h(n)')
```

脉冲响应的波形见图 2.9。

b. MATLAB 程序

```
>>x = stepseq(0, -20,120);
>>s = filter(b,a,x);
>>subplot(2,1,2);stem(n,s)
>>axis([-20,120,-.5,2.5])
>>title('阶跃响应');xlabel('n');ylabel('s(n)')
```

阶跃响应的波形见图 2.9。

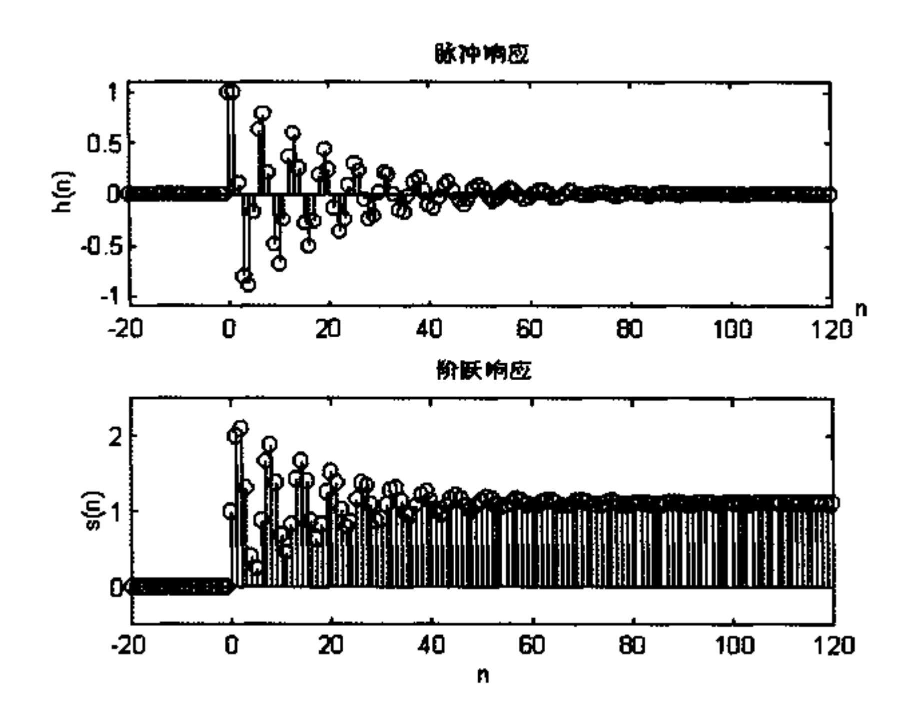


图 2.9　例 2.9 中的脉冲响应和阶跃响应

c. 为了确定系统的稳定性,我们必须对所有的 n 求出 $h(n)$。虽然我们还没有介绍一种解差分方程的方法,但可以从其脉冲响应曲线中观察到当 $n>120$ 时,$h(n)$实际上为零。因而和 $\sum|h(n)|$可以用如下 MATLAB 程序求得:

```
>>sum(abs(h))
ans = 14.8785
```

这意味着系统是稳定的。另一种方法是利用稳定条件(2.20),先用 MATLAB 的求根函数 roots:

```
>>z=roots(a);
>>magz=abs(z)
magz =0.9487
      0.9487
```

因为这两个根的模都小于一,系统是稳定的。

在前一节中,注意两个序列中只要有一个是无限长的,就不能使用 conv 函数。实际上如果有一个序列是有限长的,还是可能用 MATLAB 来对其卷积进行数值估计。这时应当用 filter 函数,这可从下面的例子中看到。

□ **例 2.10** 我们来考虑例 2.5 中给出的卷积。输入是有限序列:

$$x(n)=u(n)-u(n-10)$$

而脉冲响应是无限序列:

$$h(n)=(0.9)^n u(n)$$

求 $y(n)=x(n)*h(n)$。

解: 如果由脉冲响应 $h(n)$ 给出的 LTI 系统能用差分方程描述,则 $y(n)$ 可由 filter 函数求得。由 $h(n)$ 的表示式

$$(0.9)h(n-1)=(0.9)(0.9)^{n-1}u(n-1)=(0.9)^n u(n-1)$$

或

$$\begin{aligned}h(n)-(0.9)h(n-1)&=(0.9)^n u(n)-(0.9)^n u(n-1)\\&=(0.9)^n[u(n)-u(n-1)]=(0.9)^n\delta(n)\\&=\delta(n)\end{aligned}$$

最后一步是因为 $\delta(n)$ 只在 $n=0$ 处非零。根据定义,$h(n)$ 是输入为 $\delta(n)$ 时 LTI 系统的输出。因此可把 $\delta(n)$ 换成 $x(n)$,$h(n)$ 换成 $y(n)$,差分方程成为:

$$y(n)-0.9y(n-1)=x(n)$$

现在 MATLAB 的 filter 函数可以间接用来计算卷积了。

```
>>b=[1]; a=[1,-0.9];
>>n=-5:50; x=stepseq(0,-5,50) - stepseq(10,-5,50);
>>y=filter(b,a,x);
>>subplot(1,1,1);
>>subplot(2,1,2); stem(n,y); title('输出序列')
>>xlabel('n'); ylabel('y(n)'); axis([-5,50,-0.5,8])
```

输出的波形见图 2.10，它和图 2.6 的结果完全一样。

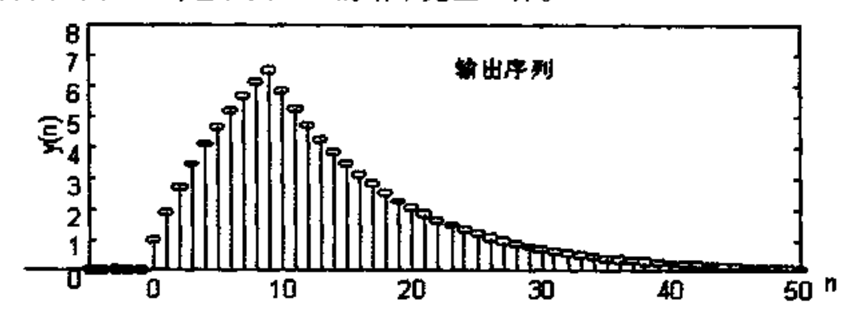


图 2.10　例 2.10 的输出序列

在例 2.10 中，脉冲响应是一个可以找到它的差分方程表达式的单边指数序列。这意味着并非所有无限长脉冲响应都能变换成差分方程的。然而上述分析可以扩展到单边指数序列的线性组合，它对应于高阶差分方程。我们将在第四章讨论这个从一种表示式转换为另一形式的论题。

零输入和零状态响应

在数字信号处理中差分方程通常是从 $n=0$ 开始按时间向前求解的。因此需要 $x(n)$ 和 $y(n)$ 的初始条件才能求得 $n\geq 0$ 处的输出。于是差分方程为：

$$y(n)=\sum_{m=0}^{M} b_m x(n-m)-\sum_{k=1}^{N} a_k y(n-k);n\geq 0 \qquad (2.21)$$

初始条件为：

$$\{y(n);-N<=n<=-1\} \quad 及 \quad \{x(n);-M<=n<=-1\}$$

(2.21)的解可写成下面的形式：

$$y(n)=yzi(n)+yzs(n)$$

其中 $yzi(n)$ 称为零输入解，它是由初始条件（假定它存在）独立造成的解；而零状态解

$yzs(n)$则是由输入 $x(n)$独立产生的解(假定初始条件为零)。在 MATLAB 中 filter 函数的另一种形式可以用来在给定初始条件下解差分方程。我们将在第四章说明这种形式的用法。

数字滤波器

滤波器是设计来进行频率选择或频率分辨任务的线性时不变系统的通用名称。因此离散时间 LTI 系统也称为数字滤波器。它可分为两类。

FIR 滤波器

如果一个 LTI 系统的单位脉冲响应长度有限,则此系统称为有限长度脉冲响应(或 FIR)滤波器。因此对一个 FIR 滤波器,在 $n<n_1$ 和 $n>n_2$时,$h(n)=0$。差分方程(2.18)中的下列部分描述了一个因果 FIR 滤波器。

$$y(n)=\sum_{m=0}^{M} b_m x(n-m) \tag{2.22}$$

而 $h(0)=b_0, h(1)=b_1, \ldots, h(M)=b_M$。而所有其他的 $h(n)$为 0。FIR 滤波器也称为非递推或滑动平均(MA)滤波器。在 MATLAB 中,FIR 滤波器既可用脉冲响应的值 $h(n)$也可用差分方程的系数$\{b_m\}$和$\{a_0=1\}$表示。因此要实现 FIR 滤波器我们既可用 conv(x,h)函数(及我们讨论过的其修改形式)或 filter(b,1,x)函数。需要注意这两种方法的输出有一个差别。由 conv(x,h)函数产生的输出的长度比 $x(n)$和 $h(n)$序列都要长。另一方面,由 filter(b,1,x)函数产生的输出序列的长度恰好与 $x(n)$序列的长度相同。实际上(特别是为处理信号),我们鼓励使用 filter 函数。

IIR 滤波器

如果一个 LTI 系统的脉冲响应具有无限长度,则此系统称为无限长脉冲响应(或 IIR)滤波器。差分方程(2.18)的下列部分:

$$\sum_{k=0}^{N} a_k y(n-k)=x(n) \tag{2.23}$$

描述了一个递推滤波器,其中输出 $y(n)$可用其以前算得的值递推计算而得,所以也称为自回归(AR)滤波器。这样的滤波器的脉冲响应是无限长的因而表示为 IIR 滤波器。通用方程(2.18)也描述了一个 IIR 滤波器。它有两个部分:一个 AR 部分和一个 MA 部分。这样的滤波器称为自回归滑动平均,或 ARMA 滤波器。在 MATLAB 中 IIR 滤波器是用差分方程的系数$\{b_m\}$和$\{a_k\}$来描述并由 filter(b,a,x)来实现的。

习　题

题 2.1　用 MATLAB 产生并画出(用 stem 函数)下列序列的样本:

a. $x_1(n)=\sum_{m=0}^{10}(m+1)[\delta(n-2m)-\delta(n-2m-1)], 0\le n\le 25$.

b. $x_2(n)=n^2[u(n+5)-u(n-6)]+10\delta(n)+20(0.5)^n[u(n-4)-u(n-10)]$.

c. $x_3(n)=(0.9)^n\cos(0.2\pi n+\pi/3),\ 0\le n\le 20$.

d. $x_4(n)=10\cos(0.0008\pi n^2)+w(n), 0\le n\le 100$,

其中 $w(n)$是一个在$[-1,1]$之间均匀分布的随机序列。问如何表征此序列?

e. $\tilde{x}_5(n)=\{\ldots,1,2,3,\underset{\uparrow}{2},1,2,3,2,1,\ldots\}_{周期的}$画出五个序列。

题 2.2 令 $x(n)=[1,-2,4,6,-\underset{\uparrow}{5},8,10]$,产生并画出下列序列的样本。

a. $x_1(n)=3x(n+2)+x(n-4)-2x(n)$

b. $x_2(n)=5x(5+n)+4x(n+4)+3x(n)$

c. $x_3(n)=x(n+4)x(n-1)+x(2-n)x(n)$

d. $x_4(n)=2e^{0.5n}x(n)+\cos(0.1\pi n)x(n+2), -10\le n\le 10$

e. $x_5(n)=\sum_{k=1}^{5} nx(n-k)$

题 2.3 复数指数序列 $e^{j\omega_0 n}$或正余弦序列是周期性的,如果其归一化频率 $f_0 \triangleq \frac{\omega_0}{2\pi}$是一个有理数,即 $f_0 \triangleq K/N$,其中 K 和 N 为整数。

a. 证明上述结果。

b. 产生并画出 $\cos(0.3\pi n), -20\leqslant n\leqslant 20$。这个序列是周期的吗? 如果它是,其基本周期是多少? 仔细研究此图后,对上述整数 K 和 N 你能给出什么解释?

c. 产生并画出 $\cos(0.3n), -20\leqslant n\leqslant 20$。这个序列是周期的吗? 从图上你能得出什么结论? 如果需要,可检查从 MATLAB 中的序列的值以便得出你的解答。

题 2.4 将题 2.2 中所给的序列分解为偶和奇分量。用 stem 函数画出这些分量。

题 2.5 复数序列被称为共轭对称的,如果

$$x_e(n)=x_e^*(-n)$$

相似地,复数序列被称为共轭反对称的,如果

$$x_o(n)=-x_o^*(-n)$$

任何复值序列 $x(n)$均可分解为:

$$x(n)=x_e(n)+x_o(n)$$

其中 $x_e(n)$和 $x_o(n)$分别等于

$$x_e(n)=[x(n)+x^*(-n)]/2 \quad 和 \quad x_o(n)=[x(n)-x^*(-n)]/2 \tag{2.24}$$

a. 修改本教材中的 evenodd 函数,使它能接受任意序列并把它分解成式(2.24)表示的对称和反对称分量。

b. 分解下列序列

$$x(n)=10e^{-(0.4\pi n)}, \quad 0\le n\le 10$$

成为共轭对称和共轭反对称的分量。画出它们的实部和虚部以验证这一分解结果(用 subplot 函数)。

题 2.6 信号的扩展(或抽取,或降低采样频率)定义为:

$$y(n)=x(nM)$$

其中 $x(n)$的采样频率被降低了整数因子 M。例如,若:

$$x(n)=\{\ldots,-2,4,\underset{\uparrow}{3},-6,5,-1,8,\ldots\}$$

把采样频率减少为整数因子 2,可得:

$$y(n)=\{\ldots,-2,\underset{\uparrow}{3},5,8,\ldots\}$$

a. 开发一个 MATLAB 函数 dnsample,其格式为:

function y = dnsample (x, M)

用以实现上述运算。在应用 MATLAB 的下标功能时要特别注意时间轴的原点 $n=0$。

b. $x(n)=\sin(0.125\pi n)$, $-50\leqslant n\leqslant 50$。频率降低因子为 4,求 $y(n)$。用 subplot 函数分别画出 $x(n)$和 $y(n)$并对结果进行讨论。

c. 用 $x(n)=\sin(0.5\pi n)$, $-50\leqslant n\leqslant 50$ 重复上题。定性地讨论降低采样频率对信号的影响。

题 2.7 求出下列序列的自相关序列 $r_{yx}(l)$和互相关序列 $rxy(l)$。

$$x(n)=(0.9)^n,\quad 0\leqslant n\leqslant 20;\quad y(n)=(0.8)^{-n};\quad -20\leqslant n\leqslant 0$$

你能观察出什么结果?

题 2.8 在某个音乐厅中,原始音频信号的回音会由于墙壁和天花板的反射而产生。听众所感受的音频信号是 $x(n)$和它的回音的合成。令:

$$y(n)=x(n)+\alpha x(n-k)$$

其中 k 是总迟延的脉冲个数而 α 是其相对强度。我们要用相关分析来估计该迟延。

a. 用解析方法由自相关 $r_{xx}(l)$确定自相关 $r_{yy}(l)$。

b. 令 $x(n)=\cos(0.2\pi n)+0.5\cos(0.6\pi n)$, $\alpha=0.1$ 及 $k=50$。产生 200 个 $y(n)$样本并求出其自相关。从观察 $r_{yy}(l)$中能否得到 α 和 k?

题 2.9 下面给出三个系统:

$$T_1[x(n)]=2^{x(n)};\quad T_2[x(n)]=3x(n)+4;\quad T_3[x(n)]=x(n)+2x(n-1)-x(n-2)$$

a. 用(2.8)式解析地确定上述系统是否是线性的。

b. 令 $x_1(n)$为在[0,1]之间均匀分布的随机序列 $0\leqslant n\leqslant 100$,又令 $x_2(n)$为一个均值为零方差为 10 的高斯随机序列 $0\leqslant n\leqslant 100$ 。用这些序列测试上述系统的线性。在(2.8)中,常数值 a_1和 a_2可任意选择。要求用上述序列的几个实现来求得答案。

题 2.10 下面给出三个系统:

$$T_1[x(n)]=\sum_{0}^{n} x(k);\quad T_2[x(n)]=\sum_{n-10}^{n+10} x(k);\quad T_3[x(n)]=x(-n)$$

a. 用(2.9)式解析地确定上述系统是否是时不变的。

b. 令 $x(n)$为一零均值方差为 10 的高斯随机序列 $0\leqslant n\leqslant 100$。利用这个序列,测试上述诸系统的时不变性。(2.9)中采样移位值 k 可任意选择。要求用上述序列的几个实现来求得答案。

题 2.11 解析地确定题 2.9 和 2.10 中所给诸系统的稳定性和因果性。

题 2.12 在(2.11)中定义的线性卷积具有若干特性:

$$\begin{aligned}
x_1(n)*x_2(n)&=x_2(n)*x_1(n) &&:\text{交换律}\\
[x_1(n)*x_2(n)]*x_3(n)&=x_1(n)*[x_2(n)*x_3(n)] &&:\text{结合律}\\
x_1(n)*[x_2(n)+x_3(n)]&=x_1(n)*x_2(n)+x_1(n)*x_3(n) &&:\text{分配律}\\
x(n)*\delta(n-n_0)&=x(n-n_0) &&:\text{同一律}
\end{aligned}\qquad(2.25)$$

a. 解析地证明这些结果。

b. 对下面三个序列,用 conv_m 函数,验证上述特性。

$$\begin{aligned}
x_1(n)&=n[u(n+10)-u(n-20)]\\
x_2(n)&=\cos(0.1\pi n)[u(n)-u(n-30)]\\
x_3(n)&=(1.2)^n[u(n+5)-u(n-10)]
\end{aligned}$$

题 2.13 若序列 $x(n)$和 $h(n)$分别具有有限长度 N_x 和 N_h,则它们的线性卷积(2.10)可用矩阵-向量乘法求得。若 $x(n)$和 $y(n)$的元素分别排成列向量 x 和 y,可以得到:

$$\mathrm{y}=\mathbf{Hx}$$

其中在 $h(n-k)$内的线性移位量按 $n=0,\ldots,N_h-1$ 逐行排列在矩阵 **H** 中。这一矩阵具有有趣的结构形式,被称为 Toeplitz 矩阵。为了研究这个矩阵,考虑如下序列:

$$x(n)=\{\underset{\uparrow}{1},2,3,4\}\quad \text{及}\quad h(n)=\{\underset{\uparrow}{3},2,1\}$$

a. 求出线性卷积 $y(n)=h(n)*x(n)$

b. 将 $x(n)$表为一个 4 乘 1 列向量 **x** 而 $y(n)$表为一个 6 乘 1 列向量 **y**。然后求出 6 乘 4 的矩阵 **H**,使能满足 $\mathrm{y}=\mathbf{Hx}$。

c. 找出矩阵 **H** 的特点。从此特点你能否给出 Toeplitz 矩阵的定义？怎样把这个定义与时不变相比较？

d. 对于矩阵的第一列和第一行，请说出其特点。

题 2.14 MATLAB 提供了一个称为 toeplitz 的函数，可根据第一行和第一列生成 Toeplitz 矩阵。

a. 用此函数和你对题 2.13 的答案，开发另一个 MATLAB 函数来执行线性卷积。此函数的规范格式应为：

```
function [y,H] = conv_tp(h,x)
% 用 Toeplitz 矩阵的线性卷积
% ————————————————
% [y,H] = conv_tp(h,x)
% y = 列向量形式的输出序列
% H = 对应于序列 h 的 Toeplitz 矩阵,因而 y = Hx
% h = 列向量形式的脉冲响应序列
% x = 列向量形式的输入序列
```

用题 2.13 中给出的序列验证你的函数。

题 2.15 令 $x(n)=(0.8)^n u(n)$

a. 解析地求 $x(n)*x(n)$。

b. 用 filter 函数求出 $x(n)*x(n)$ 的前 50 个样本。将结果与 a 部分的结果相比较。

题 2.16 一个特定的线性和时不变系统，描述它的差分方程如下：

$$y(n)-0.5y(n-1)+0.25y(n-2)=x(n)+2x(n-1)+x(n-3)$$

a. 确定系统的稳定性。

b. 在 $0\leqslant n\leqslant 100$ 之间求得并画出系统的脉冲响应，从脉冲响应确定系统的稳定性。

c. 如果此系统的输入为 $x(n)=[5+3\cos(0.2\pi n)+4\sin(0.6\pi n)]u(n)$。在 $0\leqslant n\leqslant 200$ 间求出 $y(n)$ 的响应。

题 2.17 一个简单的数字微分器表为下式：

$$y(n)=x(n)-x(n-1)$$

它计算输入信号的后向一阶差分。对下列序列执行这个微分并画出其结果。评论这个简单的微分器的可用性。

a. $x(n)=5[u(n)-u(n-20)]$：矩形脉冲

b. $x(n)=n[u(n)-u(n-10)]+(20-n)[u(n-10)-u(n-20)]$：三角脉冲

c. $x(n)=\sin(\pi n/25)[u(n)-u(n-100)]$：正弦脉冲

第三章　离散时间付利叶分析

我们已经看到一个线性时不变系统如何用它对单位脉冲序列的响应来表示。这个响应称为单位脉冲响应 $h(n)$，它使得我们能通过线性卷积计算该系统对任意输入 $x(n)$ 的响应如下：

$$x(n)\longrightarrow\boxed{h(n)}\longrightarrow y(n)=h(n)*x(n)$$

这个卷积的表示式是基于这样的事实：即任何信号可以用单位样本的倍乘和迟延的线性组合来表示。同样地，我们也可以把任意离散信号表为第二章所介绍的基本信号的线性组合。每个基本信号集提供了一种新的信号表示式。每种表示式有某些优点和某些缺点，取决于所研究的系统的类型。然而当系统是线性的和时不变的时候，只有一个表示式成为最有用的。它是基于复数指数信号集 $\{e^{j\omega n}\}$ 的并称为离散时间付利叶变换。

离散时间付利叶变换(DTFT)

如果 x(n)是绝对可加的，即 $\sum_{-\infty}^{\infty}|x(n)|<\infty$，则其离散时间付利叶变换表为：

$$X(e^{j\omega})\triangleq F[x(n)]=\sum_{n=-\infty}^{\infty}x(n)e^{-j\omega n} \tag{3.1}$$

$X(e^{j\omega})$的离散时间付利叶逆变换(IDTFT)可表示为：

$$x(n)\triangleq F^{-1}[X(e^{j\omega})]=\frac{1}{2\pi}\int_{-\pi}^{\pi}X(e^{j\omega})e^{j\omega n}d\omega \tag{3.2}$$

算子 $F[\cdot]$把一个离散信号 $x(n)$变换成一个实变量 ω 的复值连续函数 $X(e^{j\omega})$。ω 被称为数字频率，它用弧度来度量。

□**例 3.1**　确定 $x(n)=(0.5)^n u(n)$的离散时间付利叶变换。

解：序列 $x(n)$是绝对可加的，因此它的离散时间付利叶变换存在。

$$X(e^{j\omega})=\sum_{-\infty}^{\infty}x(n)e^{-j\omega n}=\sum_{0}^{\infty}(0.5)^n e^{j\omega n}$$
$$=\sum_{0}^{\infty}(0.5e^{-j\omega})^n=\frac{1}{1-0.5e^{-j\omega}}=\frac{e^{j\omega}}{e^{j\omega}-0.5}$$

□**例 3.2** 求出下列有限长度序列的离散时间付利叶变换。

$$x(n)=\{1,\underset{\uparrow}{2},3,4,5\}$$

解:用定义(3.1)

$$X(e^{j\omega})=\sum_{-\infty}^{\infty}x(n)e^{-j\omega n}=e^{j\omega}+2+3e^{-j\omega}+4e^{-j2\omega}+5e^{-j3\omega}$$

因为 $X(e^{j\omega})$是一个复值函数,必须分别画出它的模和相角(或实部和虚部)随频率 ω 变化的曲线以便用图形表现它。现在 ω 是一个从 $-\infty$ 到 $+\infty$ 的实变量,这意味着我们只能用 MATLAB 画出 $X(e^{j\omega})$函数的一部分。利用离散时间付利叶变换的两个重要特性,可以把这个域对实值序列缩小到$[0,\pi]$区间。我们将在下一节中讨论 $X(e^{j\omega})$的一些其他重要特性。

两个重要特性

我们将不加证明地叙述下面两个特性。

1.周期性: 离散时间付利叶变换 $X(e^{j\omega})$是 ω 的周期函数,其周期为 2π。

$$X(e^{j\omega})=X(e^{j[\omega+2\pi]})$$

推论:我们只需要 $X(e^{j\omega})$的一个周期(即 $\omega\in[0,2\pi]$,或 $[-\pi,\pi]$,等等)而不需要整个 $-\infty<\omega<+\infty$ 域来分析。

2.对称性:对于实值的 $x(n)$, $X(e^{j\omega})$是共轭对称的。

$$X(e^{-j\omega})=X^*(e^{j\omega})$$

或

$$\begin{aligned}
&\mathrm{Re}[X(e^{-j\omega})]=\mathrm{Re}[X(e^{j\omega})] && (偶对称)\\
&\mathrm{Im}[X(e^{-j\omega})]=-\mathrm{Im}[X(e^{j\omega})] && (奇对称)\\
&|X(e^{-j\omega})|\ \ =|X(e^{j\omega})| && (偶对称)\\
&\angle X(e^{-j\omega})\ \ =-\angle X(e^{j\omega}) && (奇对称)
\end{aligned}$$

推理:要画出 $X(e^{j\omega})$,我们现在只需 $X(e^{j\omega})$半个周期。通常在实际中是选择 $\omega\in[0,\pi]$ 的部分。

MATLAB 实现

如果 $x(n)$ 是无限长的,则不能直接用 MATLAB 来由 $x(n)$计算 $X(e^{j\omega})$,然而我们可以用它来估计表示式 $X(e^{j\omega})$在$[0,\ \pi]$频率区间的值并且画出它的模和相角(或实部和虚部)

曲线。

□例 3.3 将[0, π]分为 501 个等间隔的点，计算例 3.1 中的 $X(e^{j\omega})$并画出其模、相角、实部、虚部的曲线。

解：MATLAB 程序

```
>>w=[0:1:500]*pi/500; % [0, pi]区域分为501点
>>X=exp(j*w) ./ (exp(j*w) - 0.5*ones(1,501));
>>magX=abs(X); angX=angle(X);
>>realX=real(X); imagX=imag(X);
>>subplot(2,2,1); plot(w/pi,magX); grid
>>xlabel('以 pi 为单位的频率'); title('幅度部分'); ylabel('幅度')
>>subplot(2,2,3); plot(w/pi,angX); grid
>>xlabel('以 pi 为单位的频率'); title('相角部分'); ylabel('弧度')
>>subplot(2,2,2); plot(w/pi,realX); grid
>>xlabel('以 pi 为单位的频率'); title('实部'); ylabel('实部')
>>subplot(2,2,4); plot(w/pi,imagX); grid
>>xlabel('以 pi 为单位的频率'); title('虚部'); ylabel('虚部')
```

所得曲线见图 3.1。注意在绘图之前我们把 w 数组除以了 pi，这样频率轴是以 π 弧度为单位的以便读数。**我们强烈推荐这种标注方法。**

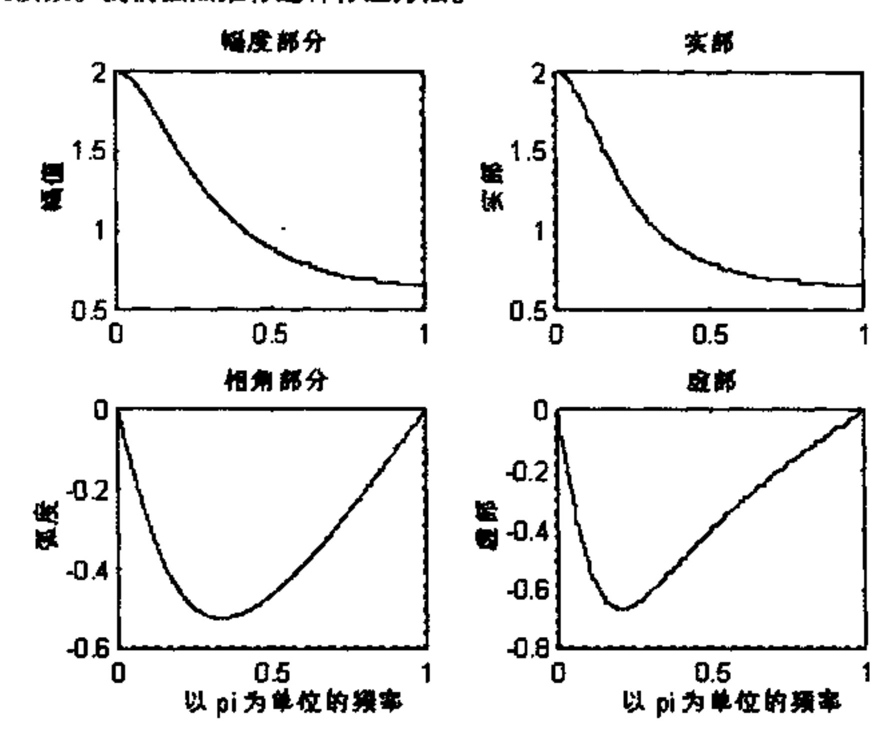


图 3.1 例 3.3 中的曲线

如果 $x(n)$是有限长的,则可以用 MATLAB 来对任意频率 ω 处的 $X(e^{j\omega})$进行数值计算。此外,如果我们是在$[0,\pi]$间按等间隔频点来估计 $X(e^{j\omega})$,则(3.1)可以用矩阵-向量相乘的运算来实现。为了弄懂这一点,我们假定序列 $x(n)$在 $n_1 \leqslant n \leqslant n_N$(即不一定在$[0,N-1]$间)有 N 个样本 ,而要估计下列诸点上的 $X(e^{j\omega})$。

$$\omega_k \triangleq \frac{\pi}{M}k, \quad k=0,1,\ ...,M$$

它们是$[0,\pi]$之间的$(M+1)$个等间隔频点。则(3.1)可写成:

$$X(e^{j\omega_k}) = \sum_{l=1}^{N} e^{-j(\pi/M)kn_l}x(n_l), \quad k=0,1,\ldots,M$$

当$\{x(n_l)\}$和$\{X(e^{j\omega_k})\}$分别排成列向量 **x** 和 **X**,我们有:

$$\mathbf{X}=\mathbf{W}\mathbf{x} \tag{3.3}$$

其中 **W** 是一个$(M+1)$乘 N 维矩阵如下:

$$\mathbf{W} \triangleq \left\{ e^{-j(\pi/M)kn_l};n_1 \leq n \leq n_N, \quad k=0,1,\ldots,M \right\}$$

此外如果我们分别将$\{k\}$和$\{n_l\}$排成列向量,则

$$\mathbf{W}=\left[\exp(-j\frac{\pi}{M}\mathbf{k}^T\mathbf{n})\right]$$

在 MATLAB 中,我们把序列和下标排成行向量,因此取(3.3)式的转置,得到:

$$\mathbf{X}^T=\mathbf{x}^T\left[\exp(-j\frac{\pi}{M}\mathbf{n}^T\mathbf{k})\right] \tag{3.4}$$

注意 $\mathbf{n}^T\mathbf{k}$ 是一个 N 乘$(M+1)$维矩阵。于是(3.4)可用 MATLAB 实现如下。

```
>> k=[0:M]; n=[n1:n2];
>> X=X*(exp(-j*pi/M)).^(n'*k);
```

□例 3.4 对例 3.2 中的 $x(n)$序列的离散时间付利叶变换,在$[0,\pi]$之间的 501 个等间隔频点上进行数值计算。

解:MATLAB 程序

```
>>n = -1:3; x = 1:5;                                % x(n)序列
>>k = 0:500; w = (pi/500) * k;                      % [0, pi] 轴分为 501 点。
>>X = x * (exp(-j * pi/500)) .^ (n' * k);           % 用矩阵-向量乘法求 DTFT
>>magX = abs(X); angX = angle(X);
>>realX = real(X); imagX = imag(X);
>>subplot(2,2,1); plot(R/500,magX); grid
>>xlabel('以 pi 为单位的频率'); title('幅度部分');
>>subplot(2,2,3); plot(R/500,angx/p);grid
>>xlabel('以 pi 为单位的频率'); title('相角部分');
>>subplot(2,2,2); plot(R/500,realX); grid
>>xlabel('以 pi 为单位的频率'); title('实部');
>>subplot(2,2,4); plot(R/500,imagX); grid
>>xlabel('以 pi 为单位的频率'); title('虚部');
```

这些频域曲线见图 3.2。注意相角曲线画成了 $-\pi$ 和 π 之间的不连续曲线,这是因为在 MATLAB 中只计算主角。

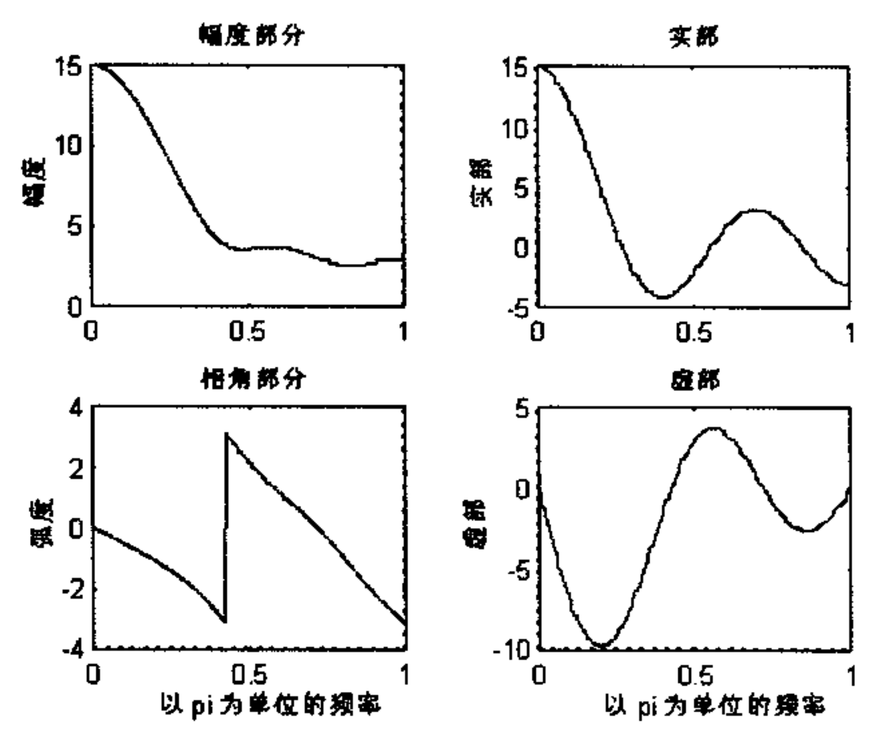


图 3.2 例 3.4 中的曲线

上述例子中的过程可以编成一个 MATLAB 函数,即 dtft 函数,以便于执行。这在习题 3.1中探讨。这个数值计算是基于定义(3.1)的,它不是计算离散时间付利叶变换的高明的方法。此外它要产生一个 N 乘($M+1$)的矩阵(3.4),当 M 和 N 很大时可能会超出学生版 MATLAB 的存储极限。在第五章中我们将详细讨论一个可计算的变换,称为离散付利叶变换(DFT),以及称为快速付利叶变换(FFT) 高效的计算方法。另外有一种办法是基于 z 变换

的。它把 MATLAB 函数 freqz 用于有限长序列。我们将在第四章中讨论。在本章中,将继续用已经讨论过的方法来计算和研究。

在下两个例子中,我们将利用复值和实值序列来研究其周期性和对称性。

□**例 3.5** 令 $x(n)=(0.9\exp(j\pi/3))^n, 0\leq n\leq 10$。求 $X(e^{j\omega})$并探讨其周期性。

解: 因为 $x(n)$是复值的,它只满足周期性。所以它被唯一地定义在一个 2π 周期上。然而,我们将在$[-2\pi,2\pi]$之间的两个周期中的 401 个频点上作计算以观察其周期性。

```
>>n=0:10; x=(0.9*exp(j*pi/3)).^n;
>>k=-200:200; w=(pi/100)*k;
>>X=x * (exp(-j*pi/100)).^(n'*k);
>>magX=abs(X); angX =angle(X);
>>subplot(2,1,1); plot(w/pi,magX);grid
>>xlabel('以 pi 为单位的频率'); ylabel('|X|')
>>title('幅度部分')
>>subplot(2,1,2); plot(w/pi,angX/pi);grid
>>xlabel('以 pi 为单位的频率'); ylabel('弧度/pi')
>>title('相角部分')
```

从图 3.3 的曲线可以看到 $X(e^{j\omega})$对 ω 是周期性的但不是共轭对称的。

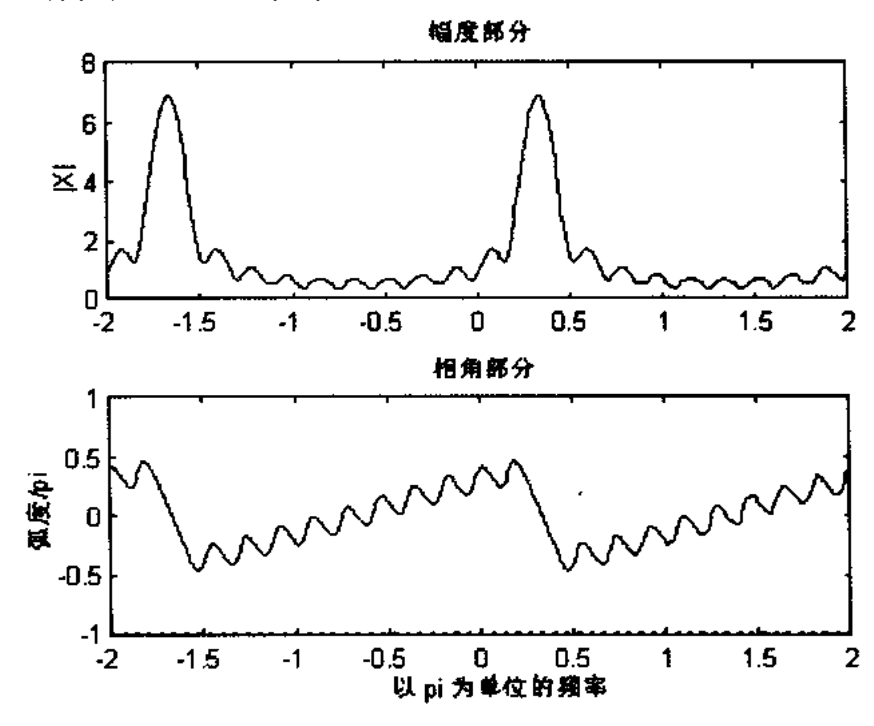


图 3.3 例 3.5 的曲线

□**例 3.6** 令 $x(n)=(-0.9)^n, -5\leq n\leq 5$,讨论其离散时间付利叶变换的共轭对称性。

解: 我们还是计算并画出 $X(e^{j\omega})$的两个周期以研究它的对称特性。

```
subplot(1,1,1)
n = -5:5; x = (-0.9).^n;
k = -200:200; w = (pi/100) * k;
X = x * (exp(-j * pi/100)) .^ (n' * k);
magX = abs(X); angX = angle(X);
subplot(2,1,1); plot(w/pi,magX);grid
axis([-2,2,0,15])
xlabel('以 pi 为单位的频率'); ylabel('|X|')
title('幅度部分')
subplot(2,1,2); plot(w/pi,angX)pi/;grid
axis([-2,2,-1,1])
xlabel('以 pi 为单位的频率'); ylabel('弧度/pi')
title('相角部分')
```

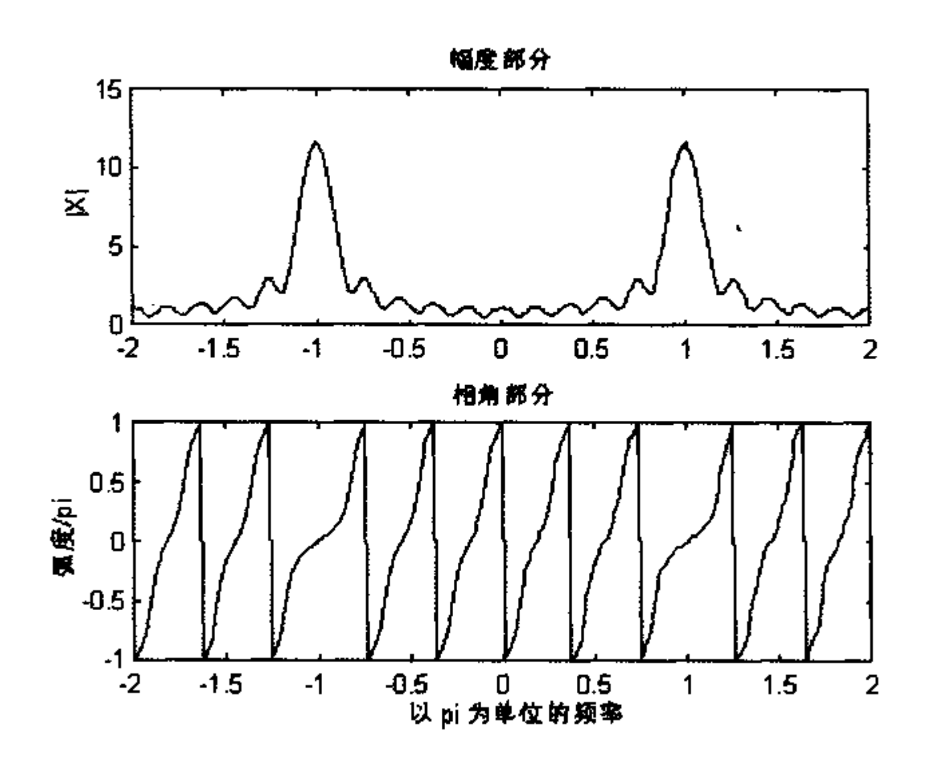


图 3.4　例 3.6 的曲线

从图 3.4 的曲线我们看到 $X(e^{j\omega})$不仅对 ω 对称,而且是共轭对称的。因此对实序列,我们只需详细画出它们从 0 到 π 之间的付利叶变换的模和相角响应。

DTFT 的特性

在上一节我们讨论了有关绘图的两个重要特性。现在来讨论其他的有用的特性,将不加证明地将它们列出如下。设 $X(e^{j\omega})$为序列 $x(n)$的离散时间付利叶变换。

1.线性: 离散时间付利叶变换是一个线性变换;即

$$F[\alpha x_1(n)+\beta x_2(n)]=\alpha F[x_1(n)]+\beta F[x_2(n)] \tag{3.5}$$

对每个 $\alpha,\beta,x_1(n)$和 $x_2(n)$。

2.时移： 时域中的移位对应于频域中的相移。

$$F[x(n-k)]=X(e^{j\omega})e^{-j\omega k} \tag{3.6}$$

3.频移： 乘以复数指数对应于一个频移。

$$F[x(n)e^{j\omega_0 n}]X(e^{j(\omega-\omega_0)}) \tag{3.7}$$

4. 共轭： 时域中的共轭对应于频移中的折叠共轭。

$$F[x^*(n)]=X^*(e^{-j\omega}) \tag{3.8}$$

5. 折叠： 时域中的折叠对应于频域中的折叠。

$$F[x(-n)]=X(e^{-j\omega}) \tag{3.9}$$

6.实序列的对称性： 我们已经研究过实序列的共轭对称性。这些实序列可以像第二章中讨论过的那样分解成它们的偶部和奇部。

$$x(n)=x_e(n)+x_o(n)$$

因而

$$\begin{aligned} F[x_e(n)]&=\mathrm{Re}[X(e^{j\omega})] \\ F[x_o(n)]&=j\mathrm{Im}[X(e^{j\omega})] \end{aligned} \tag{3.10}$$

推论： 如果序列 $x(n)$是实的和偶的，则 $X(e^{j\omega})$也是实的和偶的。因而只需画出其$[0,\pi]$之间的曲线就可完全表示清楚。

在习题 3.7 中探讨了复值序列的类似的特性。

7.卷积： 这是使系统分析易于在频域进行的最重要的特性之一。

$$F[x_1(n)*x_2(n)]=F[x_1(Zn)]F[x_2(n)]=X_1(e^{j\omega})X_2(e^{j\omega}) \tag{3.11}$$

8.乘法： 它是卷积特性的对偶。

$$F[x_1(n)\cdot x_2(n)]=F[x_1(n)]\circledast F[x_2(n)]\triangleq\frac{1}{2\pi}\int X_1(e^{j\theta})X_2(e^{j(\omega-\theta)})d\theta \quad (3.12)$$

上述类似卷积的运算称为周期卷积,用⊛来表示。它的离散形式将在第五章中讨论。

9. 能量: 序列 $x(n)$的能量可写成:

$$\varepsilon_x=\sum_{-\infty}^{\infty}|x(n)|^2=\frac{1}{2\pi}\int_{-\pi}^{\pi}|X(e^{j\omega})|^2d\omega \quad (3.13)$$

$$=\int_0^{\pi}\frac{|X(e^{j\omega})|^2}{\pi}d\omega \quad \text{(对实序列用偶对称)}$$

这就是人所共知的帕塞伐尔定理。从(3.13),$x(n)$的能量密度函数定义为:

$$\Phi_x(\omega)\triangleq\frac{|X(e^{j\omega})|^2}{\pi} \quad (3.14)$$

因而在$[\omega_1,\omega_2]$频率区间的能量可表为:

$$\int_{\omega_1}^{\omega_2}\Phi_x(\omega)d\omega,\quad 0\le\omega_1<\omega_2\le\pi$$

在以后的几个例子中,我们将用有限长序列来验证上述某些特性。我们将按照数值方法来计算每一情况的离散时间付利叶变换。虽然这并不是解析地证明上述个特性的正确性,它给我们提供了一个实验的工具。

□**例 3.7** 在本例中,我们将用实值有限长序列来验证线性特性(3.5)。令 $x1(n)$ 和 $x2(n)$为两个在 $0\le n\le10$ 区间内,平均分布于[0,1]的随机序列。然后可以用数值方法求离散时间付利叶变换,步骤如下:

```
>>x1 = rand(1,11); x2 = rand(1,11); n = 0:10;
>>alpha = 2; beta = 3;
>>k = 0:500; w = (pi/500) * k;
>>X1 = x1 * (exp(-j * pi/500)).^(n' * k);       % x1 的 DTFT
>>X2 = x2 * (exp(-j * pi/500)).^(n' * k);       % x2 的 DTFT
>>x = alpha * x1 + beta * x2;                    % x1 和 x2 的线性组合
>>X = x * (exp(-j * pi/500)).^(n' * k);          % x 的 DTFT
                                                 % 校验
>>X-check = alpha * X1 + beta * X2;              % X1 和 X2 的线性组合
>>error = max(abs(X - X-check))                  % 差值
error =
```

```
  7.1054e-015
```

因为这两个付利叶变换数组之间的最大误差不到 10^{-14},在 MATLAB 的计算精度下,这两个数组是一样的。

□例 3.8　令 $x(n)$为在 $0\leq n\leq 10$ 区间内平均分布于[0,1]之间的随机序列,并令 $y(n)=x(n-2)$。则可以检验采样移位特性(3.6)如下:

```
>>x=rand(1,11); n=0:10;
>>k=0:500; w=(pi/500)*k;
>>X=x * (exp(-j*pi/500)).^(n'*k);          % x 的 DTFT
% 信号移位两个样本点
>>y=x; m=n+2;
>>Y=y * (exp(-j*pi/500)).^(m'*k);          % y 的 DTFT
% 校验
>>Y_check=(exp(-j*2).^w).*X;               % 乘以 exp(-j2w)
>>error=max(abs(Y-Y_check))                % 差值
error=
  5.7737e-015
```

□ 例 3.9　为了检验频移特性(3.7),我们将采用图形方法。令:

$$x(n)=\cos(n\pi/2),\quad 0\leq n\leq 10 \text{ 和 } y(n)=e^{j\pi n/4}x(n)$$

然后用 MATLAB,

```
>>n=0:100; x=cos(pi*n/2);
>>k=-100:100; w=(pi/100)*k;                 %把-pi 到+pi 分为 201 点
>>X=x * (exp(-j*pi/100)).^(n'*k);           % x 的 DTFT
%
>>y=exp(j*pi*n/4).*x;                       % 把信号乘以 exp(j*pi*n/4)
>>Y=y * (exp(-j*pi/100)).^(n'*k);           % y 的 DTFT
% 图形校验
>>subplot(1,1,1)
>>subplot(2,2,1);plot(w/pi,abs(X));grid;axis([-1,1,0,60])
>>xlabel('以 pi 为单位的频率');ylabel('|X|')
>>title('X 的幅度')
>>subplot(2,2,2);plot(w/pi,angle(X)/pi);grid;axis([-1,1,-1,1])
>>xlabel('以 pi 为单位的频率'); ylabel('弧度/pi')
>>title('X 的相角')
>>subplot(2,2,3);plot(w/pi,abs(Y));grid;axis([-1,1,0,60])
```

```
>>xlabel('以 pi 为单位的频率');ylabel('|Y|')
>>title('Y 的幅度')
>>subplot(2,2,4); plot(w/pi,angle(Y)/pi); grid; axis([-1,1,-1,1])
>>xlabel('以 pi 为单位的频率'); ylabel('弧度/pi')
>>title('Y 的相角')
```

从图 3.5 可以看到，$X(e^{j\omega})$在幅度和相位图上确实都沿频率轴移位了 $\pi/4$。

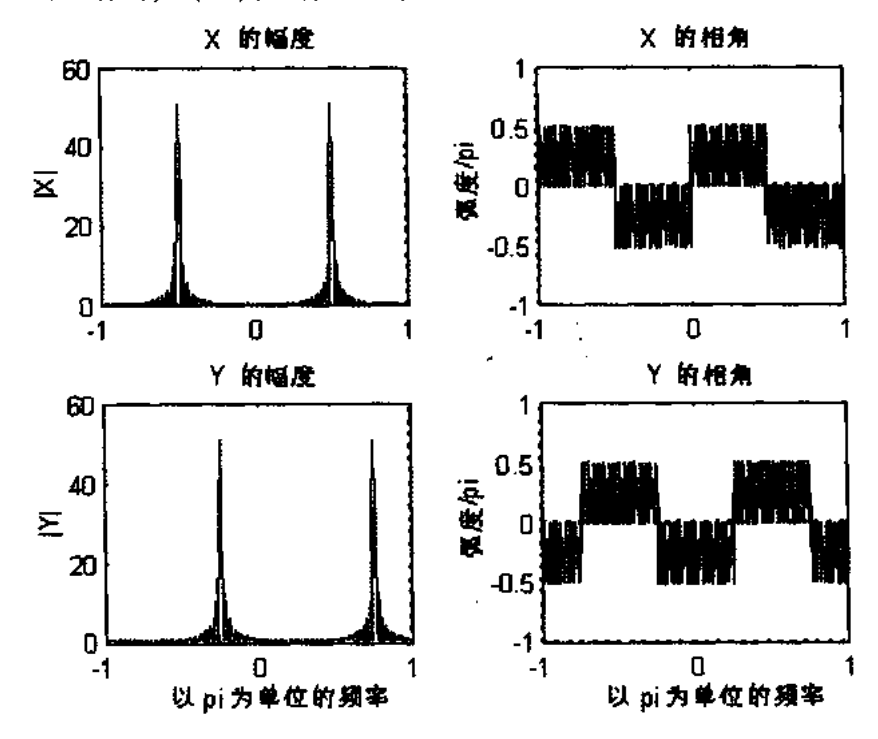


图 3.5 例 3.9 的曲线

□ **例 3.10** 为检验共轭特性(3.8)，令 $x(n)$为一复值随机序列，其长度 $-5\leq n\leq 10$，其实部和虚部都是在[0,1]之间均匀分布的。用 MATLAB 进行校验如下：

```
>>n = -5:10; x = rand(1,length(n)) + j*rand(1,length(n));
>>k = -100:100; w = (pi/100)*k;            % -pi 和 +pi 之间的频率
>>X = x * (exp(-j*pi/100)).^(n'*k);        % DTFT of x
% 共轭特性
>>y = conj(x);                              % 信号共轭
>>Y = y * (exp(-j*pi/100)).^(n'*k);        % DTFT of y
% 校验
>>Y_check = conj(fliplr(X));                % conj(X(-w))
>>error = max(abs(Y-Y_check))               % 差值
error =
    0
```

□例 3.11　为了检验折叠特性(3.9)，令 $x(n)$为在 $-5\leqslant n\leqslant 10$ 区间均匀分布于[0,1]之间的随机序列。其 MATLAB 校验程序如下：

```
>>n = -5:10; x = rand(1,length(n));
>>k = -100:100; w = (pi/100) * k;            % -pi 和 +pi 之间的频率
>>X = x * (exp(-j * pi/100)).^(n' * k);      % DTFT of x
% 折叠特性
>>y = fliplr(x); m = -fliplr(n);             % 信号折叠
>>Y = y * (exp(-j * pi/100)).^(m' * k);      % DTFT of y
% 校验
>>Y_check = fliplr(X);                       % X(-w)
>>error = max(abs(Y - Y_check))              % 差值
error =
  0
```

□例 3.12　在本例中我们检验实信号的对称特性(3.10)。令：

$$x(n)=\sin(\pi n/2),\quad -5\leqslant n\leqslant 10$$

然后用第二章中开发的 evenodd 函数，我们可以计算 $x(n)$的偶部和奇部并估计它们的离散时间付利叶变换。下面同时给出数字和图形的校验。

```
>>n = -5:10; x = sin(pi * n/2);
>>k = -100:100; w = (pi/100) * k;            % -pi 和 +pi 之间的频率
>>X = x * (exp(-j * pi/100)).^(n' * k);      % x 的 DTFT
% 信号分解
>>[xe,xo,m] = evenodd(x,n);                  % 偶部和奇部
>>XE = xe * (exp(-j * pi/100)).^(m' * k);    % xe 的 DTFT
>>XO = xo * (exp(-j * pi/100)).^(m' * k);    % xo 的 DTFT
% 校验
>>XR = real(X);                              % X 的实部
>>error1 = max(abs(XE - XR))                 % 差值
error1 =
1.8974e-19
>>XI = imag(X);                              % X 的虚部
>>error2 = max(abs(XO - j * XI))             % 差值
error2 =
  1.8033e-019
% 图形校验
>>subplot(1,1,1)
```

```
>>subplot(2,2,1); plot(w/pi,XR); grid; axis([-1,1,-2,2])
>>xlabel('以 pi 为单位的频率'); ylabel('Re(X)');
>>title('X 的实部')
>>subplot(2,2,2); plot(w/pi,XI); grid; axis([-1,1,-10,10])
>>xlabel('以 pi 为单位的频率'); ylabel('Im(X)');
>>title('X 的虚部')
>>subplot(2,2,3); plot(w/pi,real(XE)); grid; axis([-1,1,-2,2])
>>xlabel('以 pi 为单位的频率'); ylabel('XE');
>>title('偶部的变换')
>>subplot(2,2,4); plot(w/pi,imag(XO)); grid; axis([-1,1,-10,10])
>>xlabel('以 pi 为单位的频率'); ylabel('XO');
>>title('奇部的变换')
```

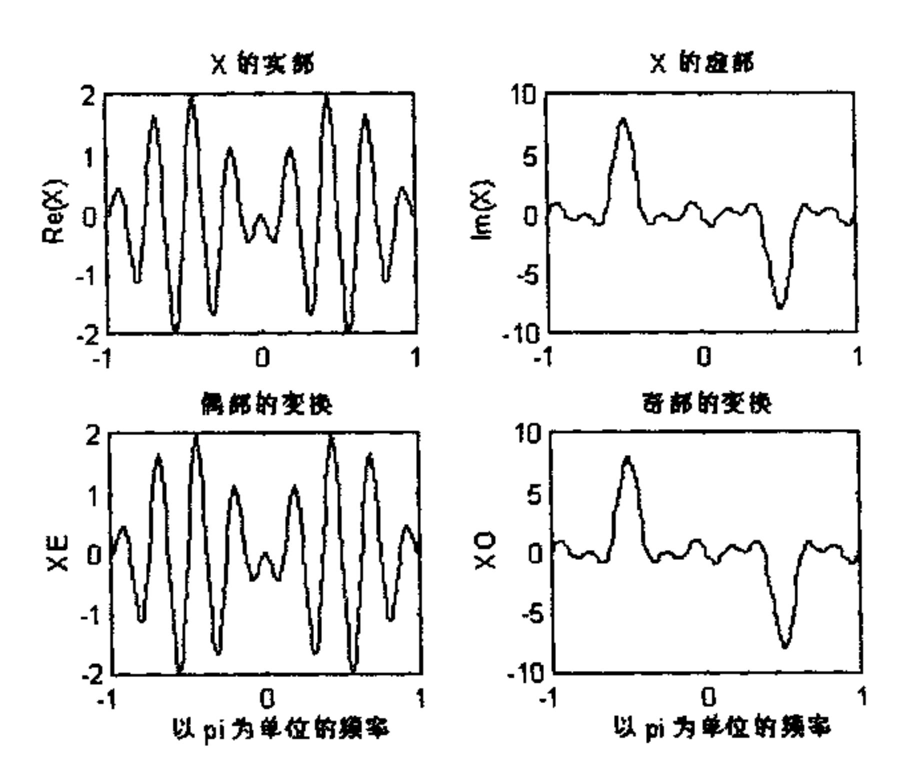


图 3.6 例 3.12 中的曲线

从图 3.6 我们看到 $X(e^{j\omega})$的实部(或 $X(e^{j\omega})$的虚部)等于 $x_e(n)$(或 $x_o(n)$)的离散时间付利叶变换。

LTI 系统的频域表示式

我们以前说过,对 LTI 系统,付利叶变换表示式是最有用的表示式。这是由于以下的原因:

对复数指数 $e^{j\omega_0 n}$的响应

令 $x(n)=e^{j\omega_0 n}$为对 LTI 系统的输入,该系统由下列脉冲响应 $h(n)$表征:

$$e^{j\omega_0 n} \longrightarrow \boxed{h(n)} \longrightarrow h(n) * e^{j\omega_0 n}$$

因而

$$\begin{aligned} y(n) = h(n) * e^{j\omega_0 n} &= \sum_{-\infty}^{\infty} h(k) e^{j\omega_0 n(n-k)} \\ &= [\sum_{-\infty}^{\infty} h(k) e^{-j\omega_0 k}] e^{j\omega_0 n} \\ &= \left[F[h(n)] \Big|_{\omega=\omega_0} \right] e^{j\omega_0 n} \end{aligned} \tag{3.15}$$

■ **定义 1　频率响应**

一个脉冲响应的离散时间付利叶变换称为一个 LTI 系统的频率响应(或传递函数)并表为:

$$H(e^{j\omega n}) \triangleq \sum_{-\infty}^{\infty} h(n) e^{-j\omega n} \tag{3.16}$$

又由(3.15),可把系统表示为:

$$x(n) = e^{j\omega_k n} \longrightarrow \boxed{H(e^{j\omega})} \longrightarrow y(n) = H(e^{j\omega_0}) \times e^{j\omega_0 n} \tag{3.17}$$

因此输出序列是输入指数序列被系统在 ω 频率处的响应修饰后的结果。这说明把 $H(e^{j\omega})$定义为频率响应是完全正确的,因为它正是复数指数输入需要乘的值以获得输出 $y(n)$。这个强有力的结果可以利用 LTI 的线性性,扩展到复数指数的线性组合中:

$$\sum_k A_k e^{j\omega_0 k} \longrightarrow \boxed{h(n)} \longrightarrow \sum_k A_k H(e^{j\omega_k n}) e^{j\omega_k n}$$

一般来说,频率响应 $H(e^{j\omega})$是 ω 的复函数。其幅度$|H(e^{j\omega})|$称作幅度(增益)响应函数,而其相角$\angle H(e^{j\omega})$则称为相位响应。我们在后面将会看到这点。

对正余弦序列的响应

令 $x(n) = A\cos(\omega_0 n + \theta_0)$为对于一个 LTI 系统 $h(n)$的输入。然后由(3.17)可以证明它的响应 $y(n)$是具有同一频率的另一个正余弦波形,其振幅放大了并有相移。即

$$y(n) = A|H(e^{j\omega_0})|\cos(\omega_0 n + \theta_0 + \angle H(e^{j\omega_0})) \tag{3.18}$$

这个响应称为稳态响应,用 $y_{ss}(n)$表示。它也能推广到正余弦序列的线性组合上去。

$$\sum_k A_k\cos(\omega_k n+\theta_k)\longrightarrow\boxed{H(e^{j\omega})}\longrightarrow\sum_k A_k|H(e^{j\omega_k})|\cos(\omega_k n+\theta_k+\angle H(e^{j\omega_k}))$$

对任意序列的响应

最后,(3.17)可以推广到任何绝对可加的序列。我们令 $X(e^{j\omega})=F[x(n)]$ 及 $Y(e^{j\omega})=F[y(n)]$;然后用卷积特性(3.11),可以得到:

$$Y(e^{j\omega})=H(e^{j\omega})X(e^{j\omega}) \tag{3.19}$$

因此,一个 LTI 系统可以在频域表示为:

$$X(e^{j\omega})\longrightarrow\boxed{H(e^{j\omega})}\longrightarrow Y(e^{j\omega})=H(e^{j\omega})X(e^{j\omega})$$

输出 $y(n)$可从 $Y(e^{j\omega})$用逆付利叶变换(3.2)算得。这需要作 MATLAB 所不擅长的积分运算。在第四章中将要见到,有另一种计算任意输入下的输出的方法。这就是用 z 变换和部分分式展开的方法。在本章中,我们集中讨论稳态响应的计算。

□**例 3.13** 求一个由 $h(n)=(0.9)^n u(n)$表征的系统的频率响应 $H(e^{j\omega})$。画出其振幅和相位响应。

解: 用(3.16)

$$H(e^{j\omega})=\sum_{-\infty}^{\infty}h(n)e^{-j\omega n}=\sum_{0}^{\infty}(0.9)^n e^{-j\omega n}$$
$$=\sum_{0}^{\infty}(0.9e^{-j\omega})^n=\frac{1}{1-0.9e^{-j\omega}}$$

因而

$$|H(e^{j\omega})|=\sqrt{\frac{1}{(1-0.9\cos\omega)^2+(0.9\sin\omega)^2}}=\frac{1}{\sqrt{1.81-1.8\cos\omega}}$$

及

$$\angle H(e^{j\omega})=-\arctan\left[\frac{0.9\sin\omega}{1-0.9\cos\omega}\right]$$

要画出这些响应,既可以导出 $|H(e^{j\omega})|$ 和 $\angle H(e^{j\omega})$ 的函数,也可导出频率响应 $H(e^{j\omega})$ 然后计算它的振幅和相位。从实用的观点来看,后一种方法更好(如(3.18)所示)。

```
>> w = [0:1:500] * pi/500; % 把[0, pi]轴分为 501 个点
```

```
>>H = exp(j * w) ./ (exp(j * w) - 0.9 * ones(1,501));
>>magH = abs(H); angH = angle(H);
>>subplot(2,1,1); plot(w/pi,magH); grid;
>>ylabel('|H|');
>>title('振幅特性');
>>subplot(2,1,2); plot(w/pi,angH/pi); grid
>>xlabel('频率(以 pi 为单位)'); ylabel('相位(单位:pi)');
>>title('相位特性');
```

所得曲线见图 3.7。

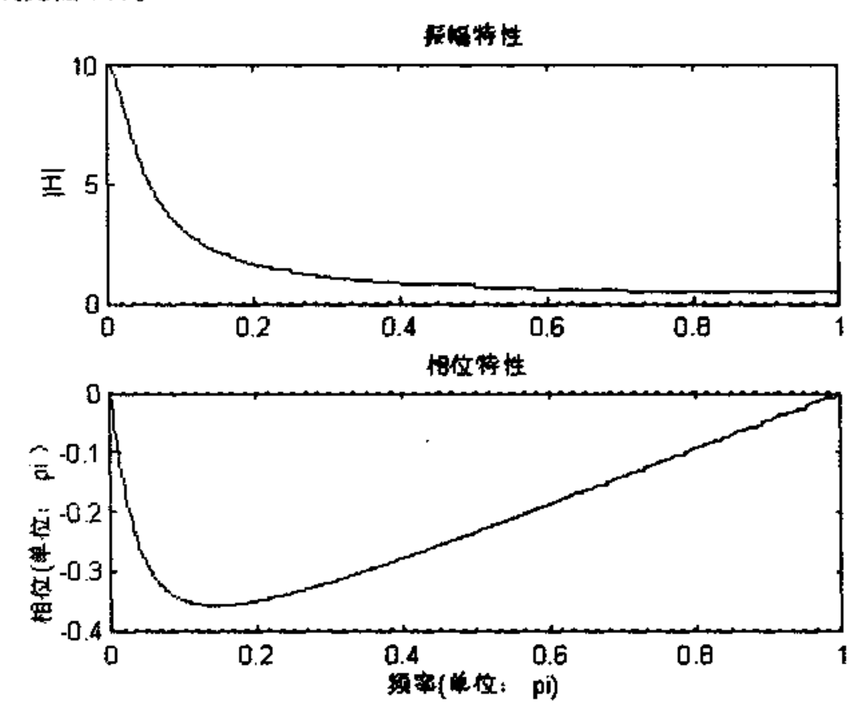


图 3.7　例 3.13 中的频率响应曲线

□**例 3.14**　令在例 3.13 中所给系统的输入为 $0.1u(n)$,求稳态输出响应 $yss(n)$。

解: 因为输入不是绝对可加的,离散时间付利叶变换在计算整个响应时不特别有用。然而可以用它来计算稳态响应。在稳态时(即 $n\to\infty$),输入是一个常数序列(或 $\omega_0=\theta_0=0$ 的正余弦序列),于是输出为:

$$y_{ss}(n)=0.1\times H(e^{j0})=0.1\times 10=1$$

其中系统在 $\omega=0$ 处的增益(也称为直流增益)为 $H(e^{j0})=10$,它可从图 3.7 中得到。

从差分方程求频率响应函数

当 LTI 系统由下列差分方程表述时

$$y(n)+\sum_{l=1}^{N}a_l y(n-l)=\sum_{m=0}^{M}b_m x(n-m) \tag{3.20}$$

为了由(3.16)求出频率响应,需要脉冲响应 $h(n)$。然而利用(3.17)可以容易地得到 $H(e^{j\omega})$。因为当 $x(n)=e^{j\omega n}$时,$y(n)$应是 $H(e^{j\omega})e^{j\omega n}$。代入(3.20),得到:

$$H(e^{j\omega})e^{j\omega n}+\sum_{l=1}^{N}a_l H(e^{j\omega})e^{j\omega(n-l)}=\sum_{m=0}^{M}b_m e^{j\omega(n-m)}$$

或在消去共同项 $e^{j\omega n}$并重新排列后得:

$$H(e^{j\omega})=\frac{\sum_{m=0}^{M}b_m e^{-j\omega m}}{1+\sum_{l=1}^{N}a_l e^{-j\omega l}} \tag{3.21}$$

只要给出差分方程系数,这个方程可以方便地在 MATLAB 中执行。

□**例 3.15** 一个 LTI 系统的差分方程如下:

$$y(n)=0.8y(n-1)+x(n)$$

a.求 $H(e^{j\omega})$。

b.求出并画出它对输入 $x(n)=\cos(0.05\pi n)u(n)$的稳态响应 $y_{ss}(n)$。

解: 把差分方程重写为 $y(n)-0.8y(n-1)=x(n)$。

a.用(3.21)式,可以得到,

$$H(e^{j\omega})=\frac{1}{1-0.8e^{-j\omega}} \tag{3.22}$$

b.系统的稳态输入为 $x(n)=\cos(0.05\pi n)$,即频率 $\omega_0=0.05\pi$,初相角 $\theta_0=0$。系统的响应为:

$$H(e^{j0.05\pi})=\frac{1}{1-0.8e^{j0.05\pi}}=4.0928e^{-j0.5377}$$

因此

$$y_{ss}(n)=4.0928\cos(0.05\pi n-0.5377)=4.0928\cos[0.05\pi(n-3.42)]$$

这意味着在输出端,正余弦函数被放大了 4.0928 倍并移位了 3.42 个采样周期。这可用 MATLAB 来验证。

```
>>subplot(1,1,1)
>>b=1; a=[1,-0.8];
>>n=[0:100];x=cos(0.05*pi*n);
>>y=filter(b,a,x);
>>subplot(2,1,1); stem(n,x);
>>xlabel('n'); ylabel('x(n)'); title('输入序列')
>>subplot(2,1,2); stem(n,y);
>>xlabel('n'); ylabel('y(n)'); title('输出序列')
```

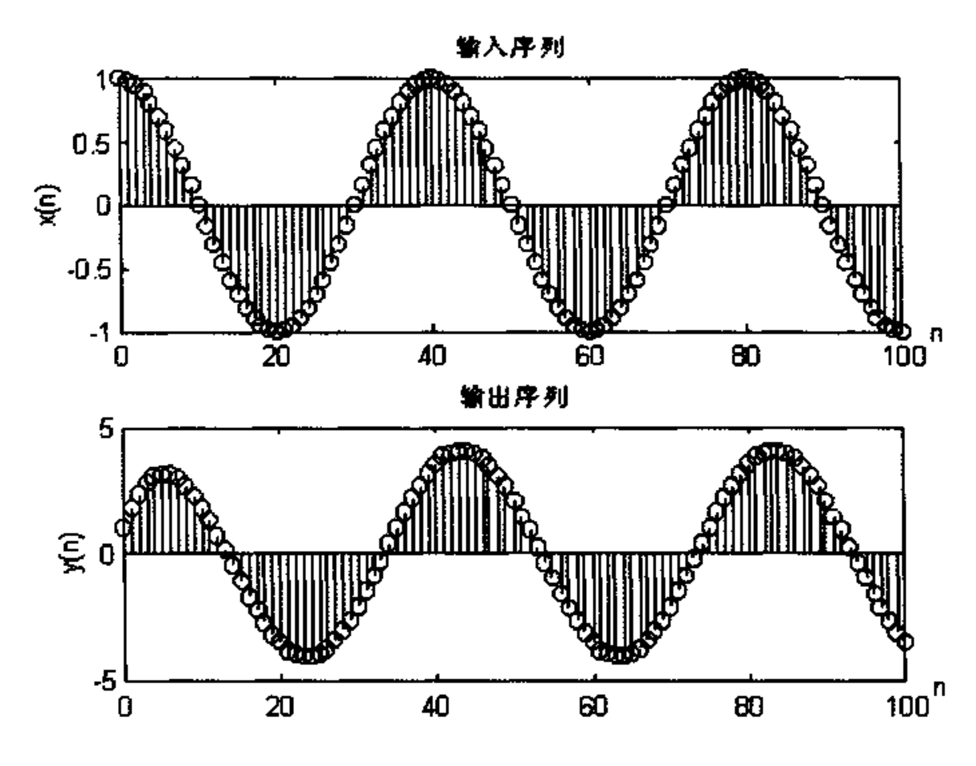


图 3.8 例 3.15 中的曲线

从图 3.8 中的曲线可以看到，$yss(n)$的振幅大约为 4。为了求出输出的移位，可以比较输入和输出的过零点。这可从图 3.8 看出，其移位大约是 3.5 个采样周期。

在例 3.15 中，系统是用一个一阶差分方程表征的。在例 3.13 中可见，用 MATLAB 来执行(3.22)是很直观的。实际中，此差分方程的阶次是很高的，因而需要一个紧凑的步骤来执行一般方程(3.21)。这可用一个简单的矩阵-向量乘法来完成。如果我们在$[0,\pi]$之间的 $k=0,1,\ldots,K$ 个等间隔频点求出 $H(e^{j\omega})$，即：

$$H(e^{j\omega_k}) = \frac{\sum_{m=0}^{M} b_m e^{-j\omega_k m}}{1+\sum_{l=1}^{N} a_l e^{-j\omega_k l}}, \quad k = 0,1,\ldots,K \tag{3.23}$$

如果令$\{b_m\}$，$\{a_l\}(a_0=1)$，$\{m=0,\ldots,M\}$，$\{l=0,\ldots,N\}$，而$\{\omega_k\}$是数组(或行向量)，

则(3.23)的分子和分母分别成为：

$$\underline{b}\exp(-j\,\underline{m}^T\underline{\omega});\quad \underline{a}\exp(-j\,\underline{l}^T\underline{\omega})$$

现在(3.23)式中的数组 $H(e^{j\omega})$ 可用 *MATLAB* 中的数组除法来计算了。给定数组$\{b_m\}$和$\{a_l\}$求频率响应函数的步骤可用一个 MATLAB 函数来实现。在例 3.16 和习题 3.15 中将进行探讨。

□**例 3.16** 一个三阶低通滤波器由以下的差分方程描述：

$$\begin{aligned}y(n)=&0.0181x(n)+0.0543x(n-1)+0.0543x(n-2)+0.0181x(n-3)\\&+1.76y(n-1)-1.1829y(n-2)+0.2781y(n-3)\end{aligned}$$

画出此滤波器的幅值和相位响应并验证它是一个低通滤波器。

解:我们用 MATLAB 来实现上述步骤并画出此滤波器的响应。

```
>>b=[0.0181,0.0543,0.0543,0.0181];         %  滤波器系数数组 b
>>a=[1.0000,-1.7600,1.1829,-0.2781];%  滤波器系数数组 a
>>m=0:length(b)-1; l=0:length(a)-1;        % 指数数组 m 和 l
>>K=500; k=1:1:K;                          % 频率指数数组 k
>>w=pi*k/K;                                % [0, pi]分成 501 个点。
>>num=b * exp(-j*m'*w);                    % 分子计算
>>den=a * exp(-j*l'*w);                    % 分母计算
>>H=num ./ den;                            % 频率响应
>>magH=abs(H); angH=angle(H);              % 幅度和相位响应
>>subplot(1,1,1);
>>subplot(2,1,1); plot(w/pi,magH); grid; axis([0,1,0,1])
>>xlabel('以 pi 为单位的频率'); ylabel('|H|');
>>title('幅度响应');
>>subplot(2,1,2); plot(w/pi,angH/pi); grid
>>xlabel('以 pi 为单位的频率'); ylabel('以 pi 弧度为单位的相位');
>>title('相位响应');
```

从图 3.9 的曲线可以看出它确实是一个低通滤波器。

模拟信号的采样和重构

在许多应用,例如在数字通信中,需要把实际世界的模拟信号,经过采样和量化运算(合在一起称为模-数转换或 ADC)变成离散信号。这些离散信号由数字信号处理器处理后,再用一个重构运算(称作数-模转换或 DAC)变成模拟信号。用付利叶分析可以从频域的观点来描述采样运算,分析其效果,然后探索重构运算。我们将假定量化的等级足够多因而离散信号的量

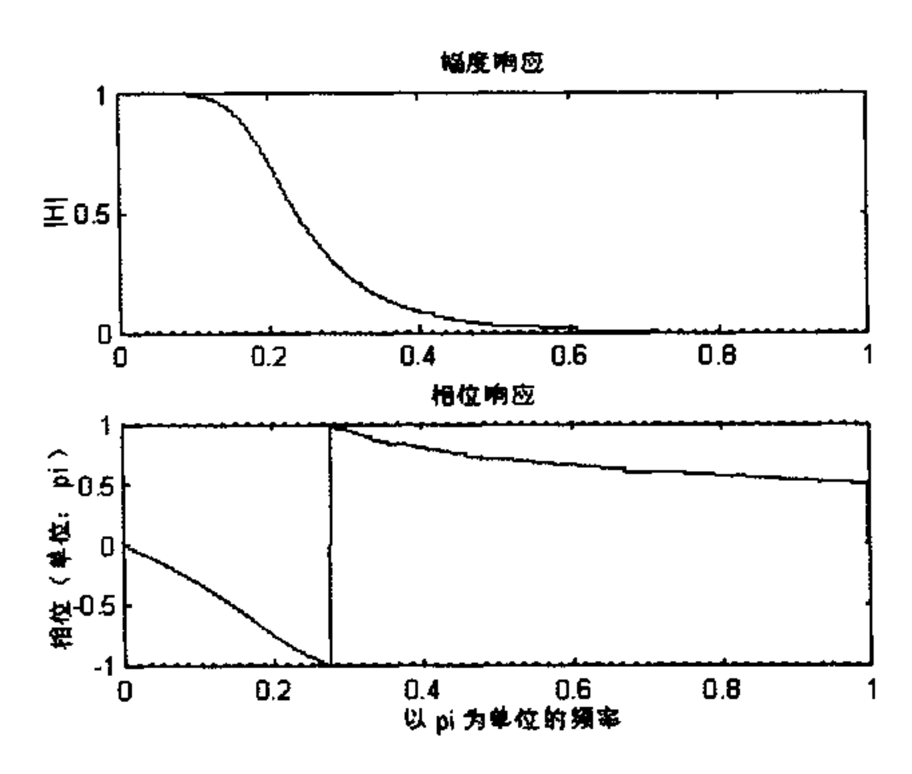


图 3.9 例 3.16 中的曲线

化效应可以忽略不计。

采样

令 $x_a(t)$为一绝对可积的模拟信号。它的连续时间付利叶变换(CTFT)表为：

$$X_a(j\Omega) \triangleq \int_{-\infty}^{\infty} x_a(t) e^{-j\Omega t} dt \tag{3.24}$$

其中 Ω 为以弧度为单位的模拟频率。逆连续付利叶变换表为：

$$x_a(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} X_a(j\Omega) e^{j\Omega t} d\Omega \tag{3.25}$$

现在以间隔 T_s 对 $x_a(t)$进行采样。以得到离散时间信号 $x(n)$：

$$x(n) \triangleq x_a(nT_s)$$

令 $X(e^{j\omega})$为 $x(n)$的离散付利叶变换。可以证明[19]，$X(e^{j\omega})$是幅度和频率都乘以系数并移相后的有限项连续付利叶变换 $X_a(j\Omega)$的和。

$$X(e^{j\omega}) = \frac{1}{T_s} \sum_{l=-\infty}^{\infty} X_a\left[j\left(\frac{\omega}{T_s} - \frac{2\pi}{T_s} l\right)\right] \tag{3.26}$$

上述关系式称为混叠公式。模拟和数字频率通过 T_s 相联系

$$\omega = \Omega T_s \tag{3.27}$$

而采样频率 F_s 为

$$F_s = \frac{1}{T_s} \quad 次/秒 \tag{3.28}$$

(3.26)式的图示见图 3.10。从中可以看到，一般的说，离散信号是相应的模拟信号的混叠形式。因为如果频谱有覆盖，高端频率会混叠到低频区去。不过如果 $X_a(j\Omega)$的无限个"复制品"互相不重叠，由付利叶变换 $X(e^{j\omega})$来恢复 $X_a(j\Omega)$(或等效地，由采样信号 $x(n)$恢复其模拟信号 $x_a(t)$)仍然是可能的。这适用于有限带宽模拟信号。

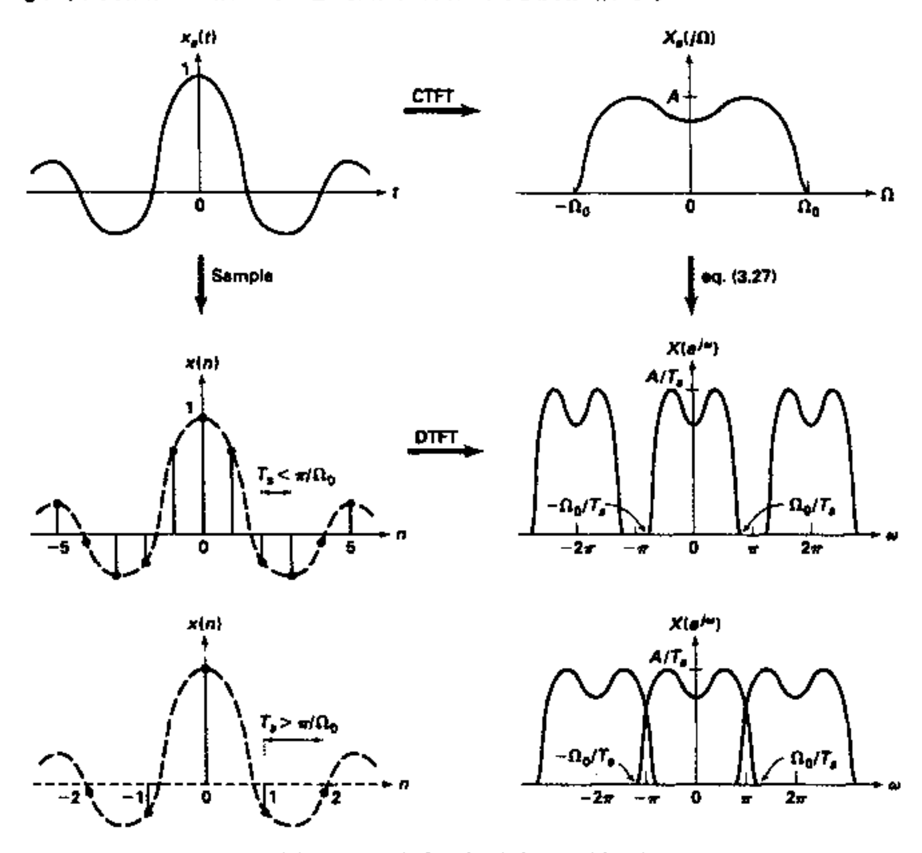


图 3.10　时域和频域中的采样运算

■ 定义 2　有限带宽信号

如果存在一个角频率 Ω_0，当 $|\Omega| > \Omega_0$ 时，$X_a(j\Omega) = 0$，则此信号是有限带宽的。频率 $F_0 = \Omega_0/2\pi$ 称为信号带宽，单位为 Hz。

参看图 3.10，如果 $\pi > \Omega_0 T_s$，或等效地 $F_s/2 > F_0$，则有：

$$X(e^{j\omega}) = \frac{1}{T_s} X\left(j\frac{\omega}{T_s}\right); \quad -\frac{\pi}{T_s} < \frac{\omega}{T_s} \le \frac{\pi}{T_s} \tag{3.29}$$

这就可导出有限带宽信号的采样定理。

■ 定理 3　采样定理

如果采样频率 F_s 大于有限带宽信号 $x_a(t)$ 带宽 F_0 的两倍,即

$$F_s > 2F_0$$

则该信号可以由它的采样值 $x(n)=x_a(nT_s)$ 重构。否则就会在 $x(n)$ 中产生混叠。对该有限带宽模拟信号的 $2F_0$ 就称为乃魁斯特频率。

必须注意,在 $x_a(t)$ 被采样以后,$x(n)$ 表示的最高模拟频率为 $F_s/2$Hz(或 $\omega=\pi$)。这和本章第一节中的离散时间付利叶变换的性质 2 的推理相吻合。

MATLAB 实现

严格地说,除了用符号处理工具箱(Symbolics)外,不可能用 MATLAB 来分析模拟信号。然而如果用时间增量足够小的很密的网格对 $x_a(t)$ 采样,就可得到一根平滑的曲线和足够长的最大时间来显示所有的模态。这样就可以进行近似分析。令 Δt 是栅网的间隔且 $\Delta t << T_s$,则

$$x_G(m) \triangleq x_a(m\Delta t) \tag{3.30}$$

可以用一个数组来仿真一个模拟信号。不要混淆采样周期 T_s 和栅网间隔 Δt,因为后者是 MATLAB 中严格地用来表示模拟信号的。类似地,付利叶变换关系(3.24)也可根据(3.30)近似为:

$$X_a(j\Omega) \approx \sum_m x_G(m)e^{-j\Omega m\Delta t}\Delta t = \Delta t \sum_m x_G(m)e^{-j\Omega m\Delta t} \tag{3.31}$$

现在,如果 $x_a(t)$(因而 $x_G(m)$)是有限长度的。则(3.31)与离散付利叶变换关系(3.3)相似,因而可以用同样的方式以 MATLAB 来实现,以便分析采样现象。

□ **例 3.17**　令 $x_a(t)=e^{-1000|t|}$,求出并绘制其付利叶变换。

解:由(3.24)

$$\begin{aligned} X_a(j\Omega) &= \int_{-\infty}^{\infty} x_a(t)e^{-j\Omega t}dt = \int_{-\infty}^{0} e^{1000t}e^{-j\Omega t}dt + \int_{0}^{\infty} e^{-1000t}e^{-j\Omega t}dt \\ &= \frac{0.002}{1+(\frac{\Omega}{1000})^2} \end{aligned} \tag{3.32}$$

因为 $x_a(t)$ 是一个实偶信号,所以它是一个实值函数。为了用数值方法估计 $X_a(j\Omega)$,必须先把 $x_a(t)$ 用一个栅格序列 $x_G(m)$ 来近似。利用 $e^{-5}\approx 0$,我们注意 $x_a(t)$ 可以用一个在

$-0.005 \leqslant t \leqslant 0.005$(或等效地[-5,5]毫秒)之间的有限长度信号来近似。类似地从(3.32),$X_a(j\Omega) \approx 0$,当 $\Omega \geqslant 2\pi(2000)$。由此选:

$$\Delta t = 5 \times 10^{-5} \ll \frac{1}{2(2000)} = 25 \times 10^{-5}$$

我们将能得到 $x_G(m)$并用 MATLAB 实现(3.31)。

```
% 模拟信号
>>Dt=0.00005;t=-0.005:Dt:0.005;xa=exp(-1000*abs(t));
% 连续时间付利叶变换
>>Wmax=2*pi*2000;K=500; k=0:1:K;W=k*Wmax/K;
>>Xa=xa * exp(-j*t'*W) * Dt;Xa=real(Xa);
>>W=[-fliplr(W), W(2:501)]; % 频率从-Wmax to Wmax
>>Xa=[fliplr(Xa), Xa(2:501)];% Xa介于-Wmax和Wmax间
>>subplot(1,1,1)
>>subplot(2,1,1);plot(t*1000,xa);
>>xlabel('t 毫秒'); ylabel('xa(t)');
>>title('模拟信号')
>>subplot(2,1,2);plot(W/(2*pi*1000),Xa*1000);
>>xlabel('频率(单位:kHz)'); ylabel('Xa(jW)*1000')
>>title('连续时间付利叶变换')
```

图3.11给出了 $x_a(t)$和 $X_a(j\Omega)$。注意为了减少计算量,我们只在$[0,4000\pi]$弧度/秒(等效地[0,2]kHz)范围内计算了 $X_a(j\Omega)$,然后将它复制到$[-4000\pi,0]$中去以便于绘图。所画出的 $X_a(j\Omega)$的图与(3.32)相符。

□**例3.18** 为了研究采样对频域各量的影响,我们用两个不同的采样频率对例3.17中的 $x_a(t)$进行采样。

a. 以 $F_s=5000$样本/秒采样 $x_a(t)$得到 $x_1(n)$。求并画出 $X_1(e^{j\omega})$。

b. 以 $F_s=1000$样本/秒采样 $x_a(t)$得到 $x_2(n)$。求并画出 $X_2(e^{j\omega})$。

解:a. 因为 $x_a(t)$的带宽是2kHz,奈魁斯特频率为4000样本/秒。它比所给的采样频率 F_s 低,因此混叠将(几乎)不存在。

```
% 模拟信号
>>Dt=0.00005;t=-0.005:Dt:0.005;xa=exp(-1000*abs(t));
% 离散时间信号
>>Ts=0.002; n=-25:1:25;x=exp(-1000*abs(n*Ts));
% 离散时间付利叶变换
>>K=500; k=0:1:K;w=pi*k/K;
>>X=x * exp(-j*n'*w);X=real(X);
```

```
>>w=[-fliplr(w), w(2:K+1)];
>>X=[fliplr(X), X(2:K+1)];
>>subplot(1,1,1)
>>subplot(2,1,1);plot(t*1000,xa);
>>xlabel('t 毫秒'); ylabel('x1(n)')
>>title('离散信号'); hold on
>>stem(n*Ts*1000,x); gtext('Ts=0.2 毫秒');hold off
>>subplot(2,1,2);plot(w/pi,X);
>>xlabel('以 pi 为单位的频率'); ylabel('X1(w)')
>>title('离散时间付利叶变换')
```

在图 3.12 的上图中,我们把离散信号 $x_1(n)$ 和 $x_a(t)$ 叠合在一起以强调采样。$X_1(e^{j\omega})$ 表明它是一个放大了($F_s=5000$ 倍)的 $X_a(j\Omega)$ 曲线。显然,不存在混叠现象。

b. 此时,$F_s=1000<4000$。因此必然会有明显的混叠出现。从图 3.13 可以看得很清楚,其中 $X_2(e^{j\omega})$ 的形状和 $X_a(j\Omega)$ 不同了,可以看出这是把互相交叠的 $X_a(j\Omega)$ 的复制品叠加的结果。

重 构

从采样定理和上述的例子可以清楚地看到,如果对有限带宽信号 $x_a(t)$ 以高于奈魁斯特的频率进行采样,就能从其采样序列 $x(n)$ 重构原模拟信号。重构可以考虑为两个步骤。

- 先把样本集转换成为一个加权脉冲串列。

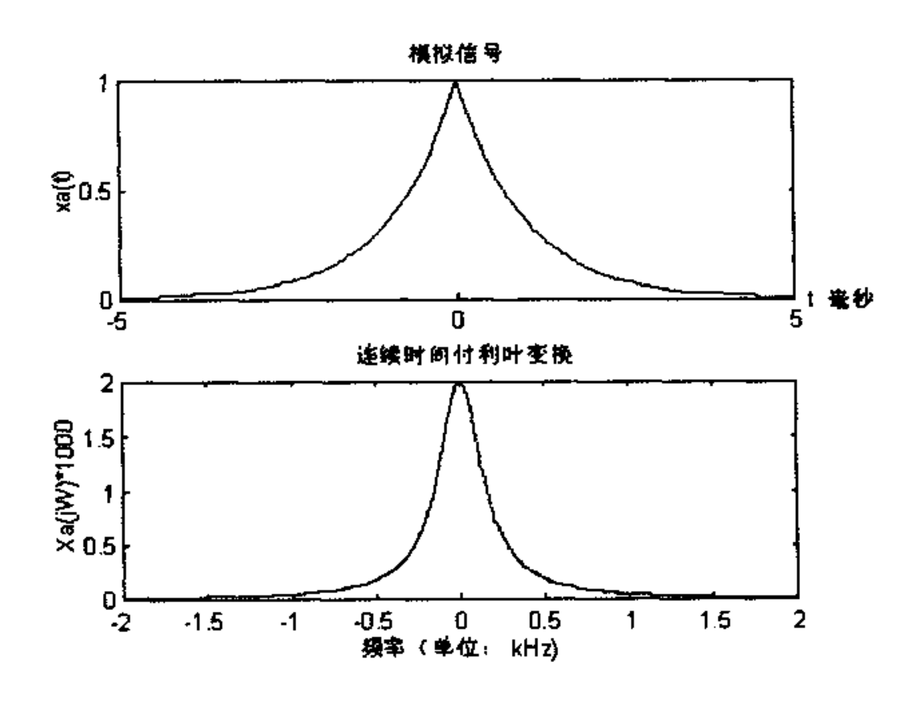


图 3.11 例 3.17 中的曲线

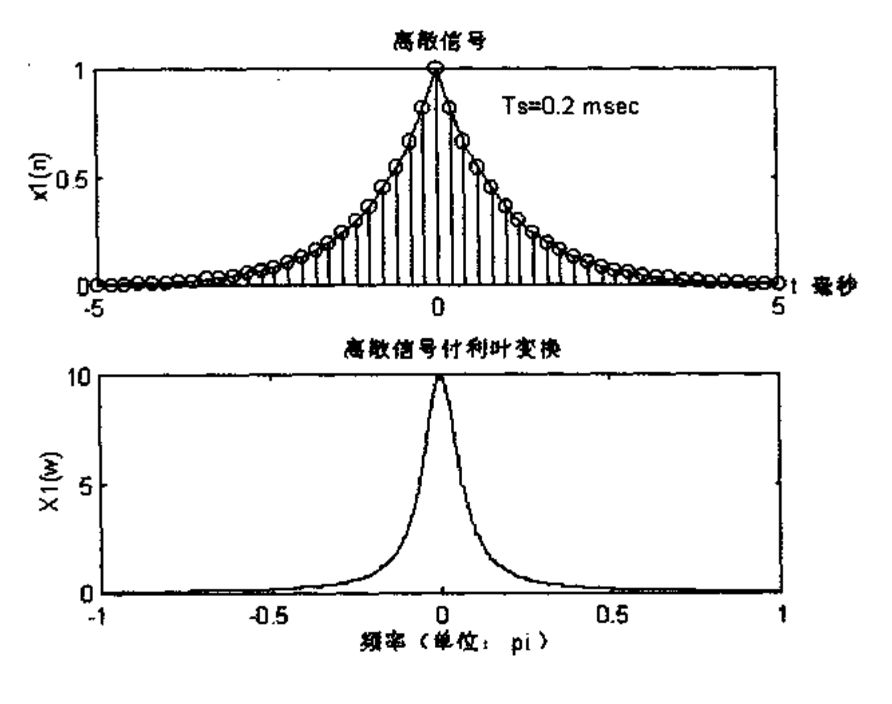


图 3.12　例 3.18a 中的曲线

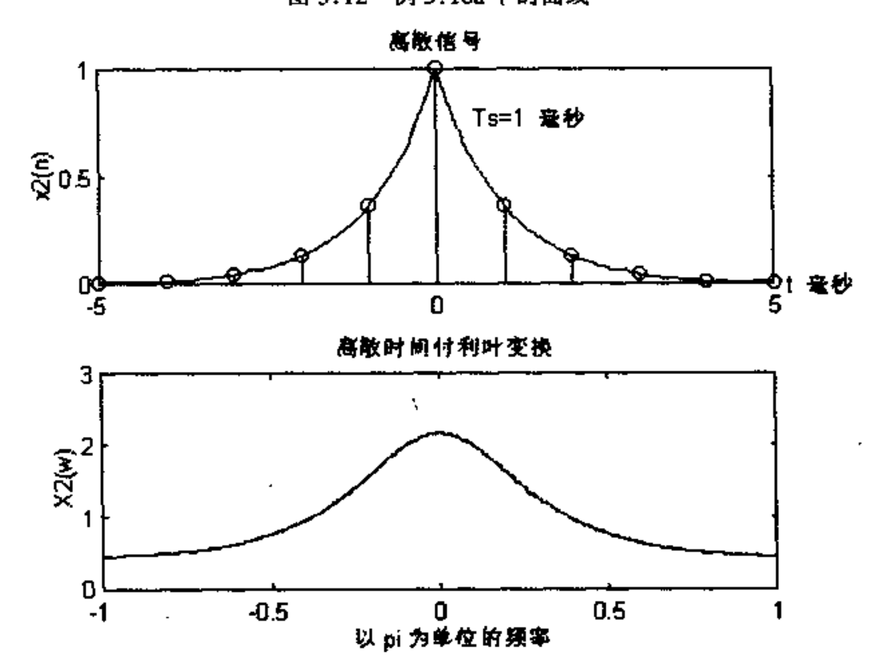


图 3.13　例 3.18b 的曲线

$$\sum_{n=-\infty}^{\infty} x(n)\delta(t-nT_s) = \ldots + x(-1)\delta(n+T_s) + x(0)\delta(t) + x(1)\delta(n-T_s) + \ldots$$

- 然后再将此脉冲串列通过一个带宽为$[-F_s/2, F_s/2]$的低通滤波器进行滤波。

$x(n)$ ⟶ [脉冲串产生] ⟶ [低通滤波器] ⟶ $x_a(t)$

这两个步骤可以用一个插值公式[19]作数学描述：

$$x_a(t)=\sum_{n=-\infty}^{\infty}x(n)\mathrm{sinc}[F_s(t-nT_s)] \tag{3.33}$$

其中 $\mathrm{sinc}(x)=\sin(\pi x)/\pi x$ 是一个内插函数。上述重构式的物理解释见图 3.14，从中可以看到，因为整个系统不是因果性的，理想的内插实际上不可能实现。

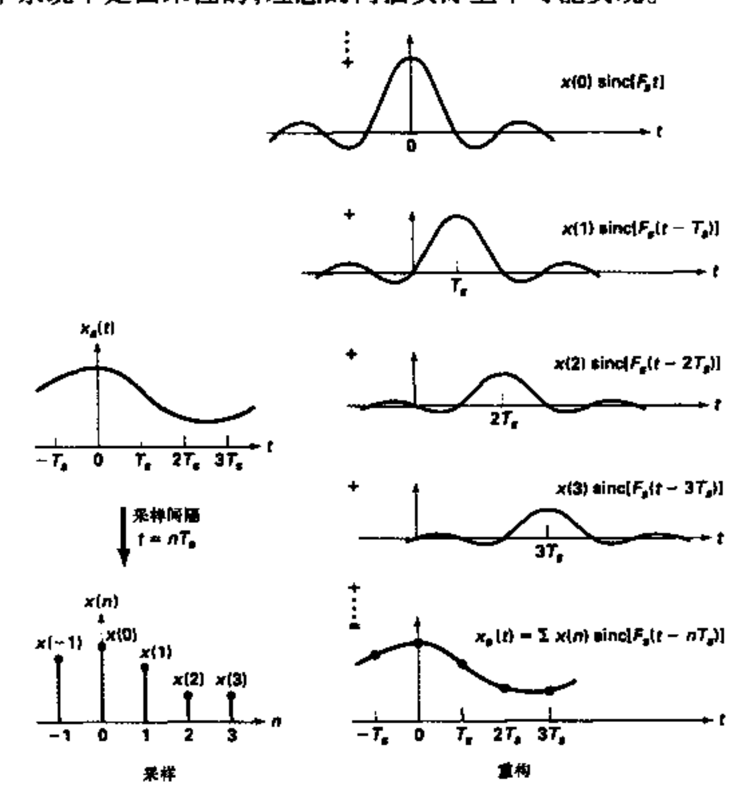


图 3.14　有限带宽信号由其采样值进行重构

实际的 D/A 变换器　实际上需要一种不同于(3.33)的方法。同样采用上述的两个步骤，但是把理想低通滤波器换成一个实际的模拟低通滤波器。(3.33)的另一个解释是：它是一个无限阶滤波器。我们只要有限阶的(实际上是低阶的)内插。这有几种方法来实现。

- **零阶保持器(ZOH)内插**：用这个内插器时，每个样本值将在整个采样周期中保持，直到收到下一个样本为止。

$$\hat{x}_a(t)=x(n),\ nT_s\le n\le(n+1)T_s$$

这可以用一个具有如下形式的内插滤波器对所得脉冲串进行滤波而得。

$$h_0(t)=\begin{cases}1,0\le t\le T_s\\0,其它\end{cases}$$

它是一个矩形脉冲。所得信号是一个分段为常数的(阶梯形的)波形,它需要一个设计良好的后接模拟滤波器,以便得到准确的波形重构。

$$x(n)\longrightarrow\boxed{\text{ZOH}}\longrightarrow\hat{x}_a(t)\longrightarrow\boxed{\text{后滤波器}}\longrightarrow x_a(t)$$

• **一阶保持器(FOH)内插**:在用这个方法时,相邻的两个样本间用直线连接。这可通过下列函数:

$$h_1(t)=\begin{cases}1+\dfrac{t}{T}, & 0<t<T_s\\ 1-\dfrac{t}{T}, & T_s\leq t\leq 2T_s\\ 0, & \text{其它}\end{cases}$$

对脉冲串进行滤波而得。同样需要一个周到设计的后接滤波器来获得准确的重构。这些内插也可推广到更高阶。在 MATLAB 中有一个特别有用的内插方法将介绍如下。

• **三次样条内插**:本方法用样条内插以获得更平滑(但不一定更精确)的对两个样本间的模拟信号的估计。这种内插不需要后接滤波器。更平滑的重构是靠采用一组称为三次样条函数的分段连续的三次多项式:

$$\begin{aligned}x_a(t)=&\alpha_0(n)+\alpha_1(n)(t-nT_s)+\alpha_2(n)(t-nT_s)^2\\&+\alpha_3(n)(t-nT_s)^3,\qquad nT_s\leq n<(n+1)T_s\end{aligned}\tag{3.34}$$

其中$\{\alpha_i(n),0\leqslant i\leqslant 3\}$是多项式系数,它们由对样本值的最小二乘分析获得(严格说来,这不是一个因果性的运算,但它是 MATLAB 中很方便的一种运算)。

MATLAB 实现

MATLAB 提供了在相邻点间内插的几种方法。函数 sinc(x)可以用来在给定有限个样本时实现(3.33)。如果给定$\{x(n),n_1\leqslant n\leqslant n_2\}$,我们要在一个很密的间距为 Δt 的栅格上内插 $x_a(t)$,则由(3.33)

$$x_a(m\Delta t)\approx\sum_{n=n_1}^{n_2}x(n)\operatorname{sinc}[F_s(m\Delta t-nT_s)],\ t_1\leq m\Delta t\leq t_2\tag{3.35}$$

这就可以用矩阵-向量的相乘运算来进行内插。程序如下:

```
>>n = n1:n2;t = t1:t2;Fs = 1/Ts;nTs = n * Ts; % Ts是采样间隔
>>xa = x * sinc(Fs * (ones(length(n),1) * t - nTs' * ones(1,length(t))));
```

注意我们不可能得到一个准确的模拟信号 $x_a(t)$,这是因为我们设定了有限数目的样本。

在下两个例题中将演示如何使用 sinc 函数并且研究时域中的混叠问题。

□ **例 3.19** 从例 3.18a 中的样本 $x_1(n)$,重构 $x_a(t)$并对结果进行评论。

解:注意 $x_1(n)$是由对 $x_a(t)$按 $T_s=1/F_s=0.0002$ 秒采样所得。我们将在 $-0.0005\leqslant t\leqslant 0.0005$ 范围内采用间距为 0.00005 的栅格作内插,它可给出 $-25\leqslant n\leqslant 25$ 间的 $x(n)$。

```
% 离散时间信号 x1(n)
>>Ts=0.0002;n=-25:1:25; nTs=n*Ts;
>>x=exp(-1000*abs(nTs));
% 模拟信号重构
>>Dt=0.00005;t=-0.005:Dt:0.005;
>>xa=x*sinc(Fs*(ones(length(n),1)*t-nTs'*ones(1,length(t))));
% 校验
>>error=max(abs(xa - exp(-1000*abs(t))))
error =
    0.0363
```

重构信号和原始模拟信号之间的最大误差为 0.0363,这是由于 $x_a(t)$不是严格的有限带宽信号,也是由于只有有限个样本。从图 3.15 可以看出,这个重构是出色的。

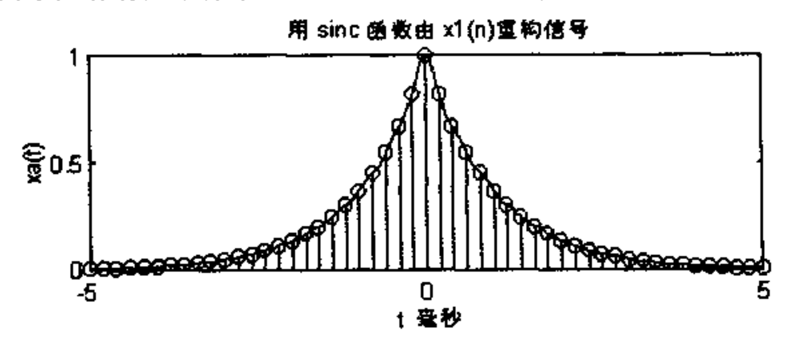


图 3.15 例 3.19 中的重构信号

□**例 3.20** 有例 3.18b 中的样本 $x_2(n)$重构 $x_a(t)$并对结果进行评论。

解:在这个情况,$x_2(n)$是由对 $x_a(t)$按 $T_s=1/F_s=0.001$ 秒采样所得。我们将在 $-0.005\leqslant t\leqslant 0.005$范围内仍采用间距为 0.00005 的栅格作内插,它可给出 $-5\leqslant n\leqslant 5$ 间的 $x(n)$。

```
% 离散时间信号 x2(n)
>>Ts=0.001; n=-5:1:5; nTs=n*Ts;
>>x=exp(-1000*abs(nTs));
% 模拟信号重构
>>Dt=0.00005;t=-0.005:Dt:0.005;
>>xa=x*sinc(Fs*(ones(length(nTs),1)*t-nTs'*ones(1,length(t))));
% 校验
>>error=max(abs(xa - exp(-1000*abs(t))))
```

```
error =
    0.1852
```

在重构信号和原始模拟信号之间的最大误差为 0.1852，这是一个明显的误差，它不能仅仅归结为 $x_a(t)$ 不是严格的有限带宽信号。从图 3.16 可以看出，在内插区域的许多地方，重构信号与原始模拟信号有显著不同。这是在时域中对混叠现象的视觉演示。

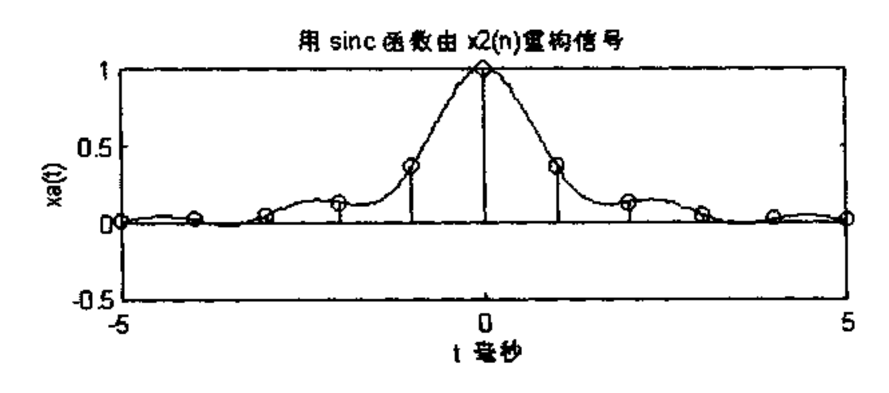


图 3.16 例 3.20 中的重构信号

信号重构的第二个 MATLAB 方法是图解法。stairs 函数在给定样本时可画出模拟信号的一个阶梯形(ZOH)的译本。而 plot 函数则给出了样本间的线性(FOH)内插。

□**例 3.21** 从例 3.18 中的样本 $x_1(n)$ 用 ZOH 和 FOH 内插画出其重构后的信号。对结果进行评论。

解：注意按题意这次重构并不需要计算 $x_a(t)$，只要用它的样本画出其曲线。

```
% 离散时间信号 x1(n) : Ts = 0.0002
> > Ts = 0.0002; n = -25:1:25; nTs = n * Ts;
> > x = exp( - 1000 * abs(nTs));
% 利用 staits 函数重构模拟信号
> > subplot(2,1,1); stairs(nTs * 1000,x);
> > xlabel('t 毫秒'); ylabel('xa(t)')
> > title('利用零阶保持器重构信号'); hold on
> > stem(n * Ts * 1000,x); hold off
%
% 利用 plot 函数重构模拟信号
> > subplot(2,1,2); plot(nTs * 1000,x);
> > xlabel('t 毫秒'); ylabel('xa(t)')
> > title('利用一阶保持器重构信号'); hold on
> > stem(n * Ts * 1000,x); hold off
```

所得曲线见图 3.17，从中可以看出 ZOH 重构的信号比较粗糙，因而对模拟信号必须作进一步的处理。FOH 重构比较好，但在 $t = 0$ 附近作仔细观察可以发现，该信号的峰值没有被准

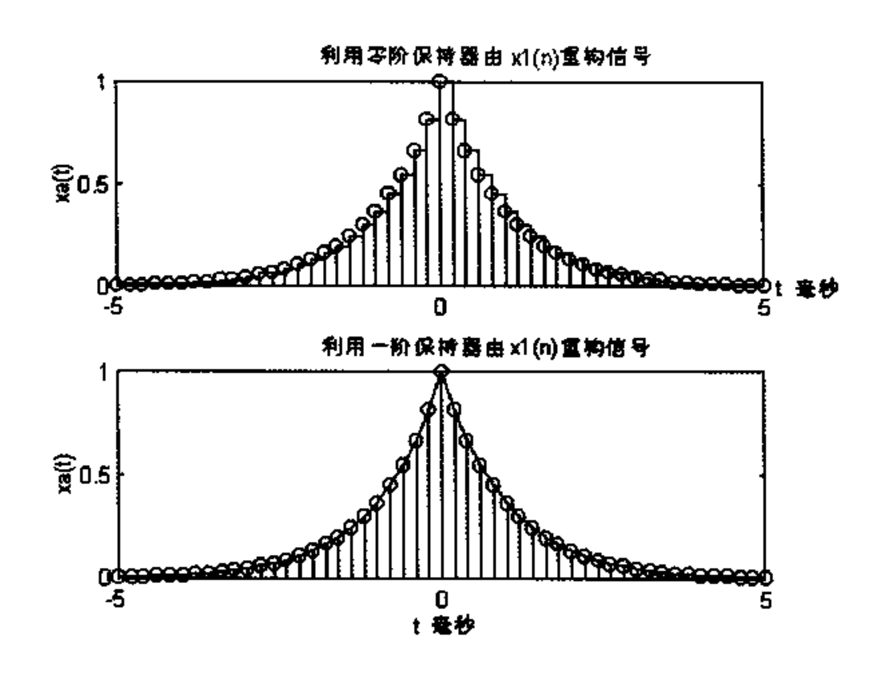


图 3.17　例 3.21 中的信号重构

确地重现。一般说来，只要采样频率比奈魁斯特频率高的多，FOH 内插给出了可以接受的重构信号。

在 MATLAB 中重构信号的第三种方法是用三次样条函数。spline 函数实现了样本点之间的内插。它由 xa = spline(nTs, x, t)调用，其中 x 和 nTs 分别是包含 nTs 处样本值 $x(n)$ 的两个数组，而 t 数组则包含一个更密的栅格，我们希望得到其上的信号 $x_a(t)$。再提醒一遍，不可能得到完全准确的模拟信号 $x_a(t)$。

□**例 3.22**　从例 3.18 中的样本 $x_1(n)$ 和 $x_2(n)$，用 spline 函数重构 $x_a(t)$，讨论其结果。

解：这个例题同例 3.18 和 3.19 相仿，因而采样参数选择同前。

```
% a)离散时间信号 x1(n): Ts = 0.0002
> > Ts = 0.0002; n = -25:1:25; nTs = n * Ts;
> > x = exp( - 1000 * abs(nTs));
% 模拟信号重构
> > Dt = 0.00005; t = -0.005:Dt:0.005;
> > xa = spline(nTs, x, t);
% check
> > error = max(abs(xa - exp( - 1000 * abs(t))))
error = 0.0317
```

在重构信号和原始模拟信号之间的最大误差为 0.0317，这是由于 $x_a(t)$ 不是严格的有限带宽信号以及非理想的内插。把这个误差与 sinc(即理想的)内插相比较，它要低一些。理想的内插更多地受到时间有限(或样本数有限)的影响。从图 3.18 的上图可以看出，视觉上这个重构是出色的。

```
% 离散时间信号 x2(n): Ts = 0.001
>> Ts = 0.001; n = -5:1:5; nTs = n * Ts;
>> x = exp( -1000 * abs(nTs));
%模拟信号重构
>> Dt = 0.00005; t = -0.005:Dt:0.005;
>> xa = spline(nTs, x, t);
% check
>> error = max(abs(xa - exp( -1000 * abs(t))))
error = 0.1679
```

在重构信号和原始模拟信号之间的最大误差为 0.1679，这是一个明显的误差，它不能仅仅归结为内插的非理想和 $x_a(t)$ 的非有限带宽。从图 3.18 下图可以看出，在内插区域的许多地方，重构信号与原始模拟信号有显著的不同。

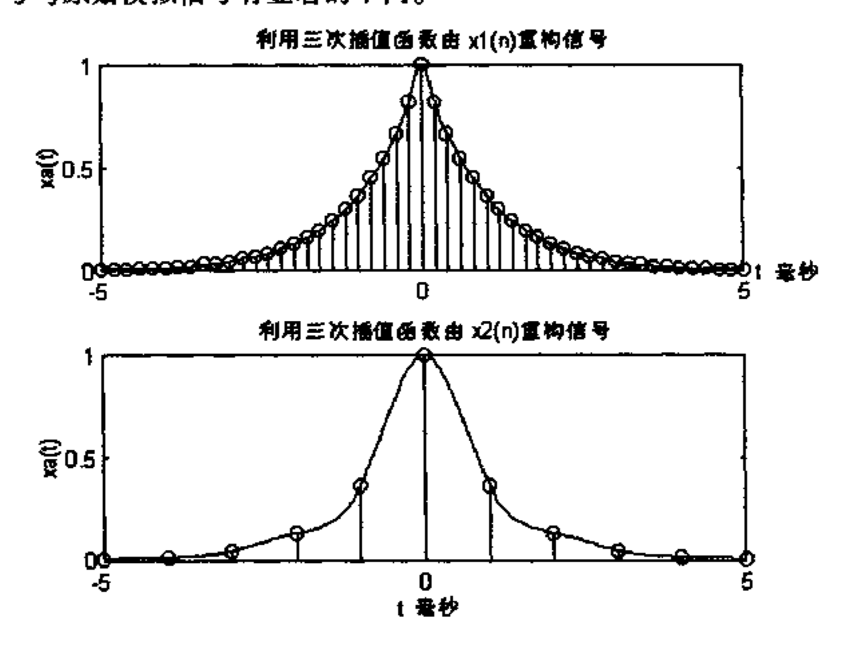


图 3.18　例 3.22 中的重构信号

从这些例子可以清楚地看出：为实用的目的，spline 内插给出了最好的结果。

习　题

题 3.1　写出一个 MATLAB 函数以计算一个有限长序列的 DTFT。该函数的格式应为：

```
function [X] = dtft(x, n, w)
% 计算离散时间付利叶变换
% [X] = dtft(x, n, w)
% X = 在 w 频率点上的 DTFT 数组
```

```
% x = n 点有限长度序列
% n = 样本位置向量
% w = 频率点位置向量
```

利用这个函数来计算以后习题中遇到的 DTFT。

题 3.2 对以下各个序列,求出其 DTFT $X(e^{j\omega})$。画出 $X(e^{j\omega})$的幅值和相位曲线。

a. $x(n)=2(0.8)^n[u(n)-u(n-20)]$

b. $x(n)=n(0.9)^n[u(n)-u(50)]$

c. $x(n)=\{\underset{\uparrow}{4},3,2,1,2,3,4\}$. 评论其相角图。

d. $x(n)=\{\underset{\uparrow}{4},3,2,1,1,2,3,4\}$. 评论其相角图。

e. $x(n)=\{\underset{\uparrow}{4},3,2,1,0,-1,-2,-3,-4\}$. 评论其相角图。

f. $x(n)=\{\underset{\uparrow}{4},3,2,1,-1,-2,-3,-4\}$. 评论其相角图。

题 3.3 用解析法求出以下各序列的 DTFT。用 MATLAB 画出 $X(e^{j\omega})$的幅值和相角曲线。

a. $x(n)=3(0.9)^3u(n)$

b. $x(n)=2(0.8)^{n+2}u(n-2)$

c. $x(n)=n(0.5)^nu(n)$

d. $x(n)=(n+2)(-0.7)^{n-1}u(n-2)$

e. $x(n)=5(-0.9)^n\cos(0.1\pi n)u(n)$

题 3.4 一个对称矩形脉冲表为:

$$R_N(n)=\begin{cases}1, & -N\le n\le N\\ 0, & \text{其它}\end{cases}$$

求出当 $N=5,15,25,100$ 时的 DTFT。对 DTFT 乘以因子使 $X(e^{j0})=1$。在$[-\pi,\pi]$区间画出归一化的 DTFT。研究这些曲线并评论它们随 N 变化的关系。

题 3.5 一对称三角脉冲表为:

$$T_N(n)=\left[1-\frac{|n|}{N}\right]R_N(n)$$

重复题 3.4 所作的工作。

题 3.6 一对称上升余弦脉冲表为:

$$C_N(n)=\left[0.5+0.5\cos\left(\frac{\pi n}{N}\right)\right]R_N(n)$$

重复题 3.4 所作的工作。

题 3.7 如第二章讨论的那样,一复值序列 $x(n)$可以分解为一个共轭对称部分 $x_e(n)$和一个共轭反对称部分 $x_o(n)$。试证明:

$$F[x_e(n)] = X_R(e^{j\omega}) \quad 及 \quad F[x_o(n)] = jX_I(e^{j\omega})$$

其中 $X_R(e^{j\omega})$ 和 $X_I(e^{j\omega})$分别是其 DTFT$X(e^{j\omega})$的实部和虚部。用第二章中开发的 MATLAB 函数对

$$x(n) = e^{j0.1\pi n}[u(n) - u(n-20)]$$

校验这个特性。

题3.8 一个复值 DTFT 也可分解为它的共轭对称部分 $X_e(e^{j\omega})$和共轭反对称部分 $X_o(e^{j\omega})$。也即：

$$X(e^{j\omega}) = X_e(e^{j\omega}) + X_o(e^{j\omega})$$

其中

$$X_e(e^{j\omega}) = \frac{1}{2}[X(e^{j\omega}) + X^*(e^{-j\omega})]及\ X_o(e^{j\omega}) = \frac{1}{2}[X(e^{j\omega}) - X^*(e^{-j\omega})]$$

试证明

$$F^{-1}[X_e(e^{j\omega})] = x_R(n)及\ F^{-1}[X_o(e^{j\omega})] = x_I(n)$$

其中 $x_R(n)$和 $x_I(n)$是 $x(n)$的实部和虚部。用第二章中开发的 MATLAB 函数对

$$x(n) = e^{j0.1\pi n}[u(n) - u(n-20)]$$

校验这个特性。

题3.9 利用频率移位特性，证明正弦脉冲

$$x(n) = \cos(\omega_0 n)R_N(n)$$

(其中 $R_N(n)$是题3.4中所给的矩形脉冲)的 DTFT 可表为：

$$X(e^{j\omega}) = \frac{1}{2}\left[\frac{\sin\{(\omega-\omega_0)N/2\}}{\sin\{(\omega-\omega_0)/2\}}\right] + \frac{1}{2}\left[\frac{\sin\{(\omega+\omega_0)N/2\}}{\sin\{(\omega+\omega_0)/2\}}\right]$$

对 $\omega_0 = \pi/2$ 及 $N = 5, 15, 25, 100$，计算并画出 $X(e^{j\omega})$。绘图区间为$[-\pi, \pi]$。评论其结果。

题3.10 令 $x(n) = T_{10}(n)$为题3.5中给出的三角脉冲。利用 DTFT 的特性，求并画出下列序列的 DTFT。

a. $x(n) = T_{10}(-n)$

b. $x(n) = T_{10}(n) - T_{10}(n-10)$

c. $x(n) = T_{10}(n) * T_{10}(-n)$

d. $x(n) = T_{10}(n)e^{j\pi n}$

e. $x(n) = T_{10}(n) \cdot T_{10}(n)$

题 3.11 对下列各脉冲响应所描述的线性时不变系统，求出其频率响应函数并画出其幅度响应 $|H(e^{j\omega})|$ 和相位响应 $\angle H(e^{j\omega})$。

a. $h(n) = (0.9)^{|n|}$

b. $h(n) = \text{sinc}(0.2n)[u(n+20) - u(n-20)]$，其中 $\text{sinc}(0) = 1$.

c. $h(n) = \text{sinc}(0.2n)[u(n) - u(n-40)]$

d. $h(n) = [(0.5)^n + (0.4)^n]u(n)$

e. $h(n) = (0.5)^{|n|}\cos(0.1\pi n)$

题 3.12 令 $x(n) = 3\cos(0.5\pi n + 60°) + 2\sin(0.3\pi n)$ 为题 3.11 中各个系统的表述输入。对每一种情况求出其输出 $y(n)$。

题 3.13 一理想低通滤波器在频域中的表述如下：

$$H_d(e^{j\omega}) = \begin{cases} 1 \cdot e^{-j\alpha\omega}, & |\omega| \le \omega_c \\ 0, & \omega_c < |\omega| \le \pi \end{cases}$$

其中 ω_c 称为截止频率而 α 称为相位滞后。

a. 用 IDTFT 关系式(3.2)求出理想脉冲响应。

b. 求并画出截断了的脉冲响应

$$h(n) = \begin{cases} h_d(n), & 0 \le n \le N-1 \\ 0, & \text{其它} \end{cases}$$

设 $N = 41$, $\alpha = 20$ 及 $\omega_c = 0.5\pi$。

c. 求并画出频率响应函数 $H(e^{j\omega})$，并将它与理想低通滤波器响应 $H_d(e^{j\omega})$ 相比。评论所得结果。

题 3.14 一理想高通滤波器在频域描述如下：

$$H_d(e^{j\omega}) = \begin{cases} 1 \cdot e^{-j\alpha\omega}, & \omega_c < |\omega| \le \pi \\ 0, & |\omega| \le \omega_c \end{cases}$$

其中 ω_c 称为截止频率而 α 称为相位滞后。

a. 用 IDTFT 关系式(3.2)求出理想脉冲响应。

b. 求并画出截断了的脉冲响应

$$h(n) = \begin{cases} h_d(n), & 0 \le n \le N-1 \\ 0, & \text{其它} \end{cases}$$

设 $N=31$，$\alpha=15$ 及 $\omega_c=0.5\pi$。

c. 求并画出频率响应函数 $H(e^{j\omega})$，并将它与理想高通滤波器响应 $H_d(e^{j\omega})$ 相比。评论所得结果。

题 3.15 一个线性时不变系统用下列差分方程描述：

$$y(n)=\sum_{m=0}^{M} b_m x(n-m)-\sum_{l=1}^{N} a_l y(n-l)$$

其频率响应函数为：

$$H(e^{j\omega})=\frac{\sum_{m=0}^{M} b_m e^{-j\omega m}}{1+\sum_{l=1}^{N} a_l e^{-j\omega l}}$$

写出一个 MATLAB 函数 freqresp 来实现上述关系。此函数的格式应为：

```
function [H] = freqresp(b,a,w)
% 由差分方程求频率响应函数
% [H] = freqresp(b,a,w)
% H = 在 w 频率点上求得的频率响应数组
% b = 分子系数数组
% a = 分母系数数组
% w = 频率点位置向量
```

题 3.16 对下列每个系统，求出其 $H(e^{j\omega})$ 并画出其幅度和相位曲线。

a. $y(n)=\sum_{m=0}^{6} x(n-m)$

b. $y(n)=x(n)+2x(n-1)+x(n-2)-0.5y(n-1)-0.25y(n-2)$

c. $y(n)=2x(n)+x(n-1)-0.25y(n-1)+0.25y(n-2)$

d. $y(n)=x(n)+x(n-2)-0.81y(n-2)$

e. $y(n)=x(n)-\sum_{l=1}^{5}(0.5)^l y(n-l)$

题 3.17 一个线性时不变系统由下列差分方程描述：

$$y(n)=\sum_{m=0}^{3} x(n-2m)-\sum_{l=1}^{3}(0.81)^l y(n-2l)$$

求系统对以下输入的稳态响应：

a. $x(n)=5+10(-1)^n$

b. $x(n)=1+\cos(0.5\pi n+\pi/2)$

c. $x(n)=2\sin(\pi n/4)+3\cos(3\pi n/4)$

d. $x(n)=\sum_{k=0}^{5}(k+1)\cos(\pi k n/4)$

e. $x(n)=\cos(\pi n)$

在每个情况产生 $x(n)$,$0 \leq n \leq 200$ 并通过 filter 函数作处理以得到 $y(n)$,把所得 $y(n)$ 与每种情况的稳态响应进行比较。

题 3.18 一个模拟信号 $x_a(t) = \sin(1000\pi t)$用以下的脉冲间隔进行采样。对每种情况画出所得离散时间信号的频谱。

a. $T_s = 0.1\text{ms}$

b. $T_s = 1\text{ms}$

c. $T_s = 0.01\text{sec}$

题 3.19 有下列模拟滤波器,它是用离散滤波器实现的。

$$x_a(t) \longrightarrow \boxed{A/D} \xrightarrow{x(n)} \boxed{h(n)} \xrightarrow{y(n)} \boxed{D/A} \longrightarrow y_a(t)$$

A/D 和 D/A 的采样频率为 100 样本/秒,而其脉冲响应为 $h(n) = (0.5)n\ u(n)$。

a. 如果 $x_a(t) = 3\cos(20\pi t)$, $x(n)$的数字频率是什么?

b. 如果 $x_a(t) = 3\cos(20\pi t)$,求 $y_a(t)$的稳态输出。

c. 如果 $x_a(t) = 3u(t)$,求 $y_a(t)$的稳态输出。

d. 求两个具有不同模拟频率的其他模拟信号,它们能给出与 $x_a(t) = 3\cos(20\pi t)$所产生的相同的稳态输出 $y_a(t)$。

e. 为了避免混叠,需要用一个预滤波器在 $x_a(t)$经过 A/D 变换器之前先予处理。对给定的框图,必须用什么样的滤波器?它的最大截止频率是多少?

题 3.20 考虑模拟信号 $x_a(t) = \sin(20\pi t)$, $0 \leq t \leq 1$。分别用 $T_s = 0.01, 0.05$, 和 0.1 秒的采样间隔对它采样以获得 $x(n)$。

a. 对每个 T_s,画出 $x(n)$。

b. 用 sinc 内插(取 $\Delta t = 0.001$),由 $x(n)$样本集重构模拟信号 $y_a(t)$,并从图中求出$y_a(t)$的频率(忽略尾部效应)。

c. 用三次样条内插,由 $x(n)$样本集重构模拟信号 $y_a(t)$,并从图中求出 $y_a(t)$的频率(忽略尾部效应)。

d. 讨论所得结果。

题 3.21 考虑模拟信号 $x_a(t) = \sin(20\pi t + \pi/4)$, $0 \leq t \leq 1$。采样间隔为 0.05 秒,得到 $x(n)$。

a. 画出 $x_a(t)$并用 plot(n,x,'o')将 $x(n)$叠加在它上面。

b. 用 sinc 内插(取 $\Delta t = 0.001$),由 $x(n)$样本集重构模拟信号 $ya(t)$,并将 $x(n)$叠加在它上面。

c. 用三次样条内插,由 $x(n)$样本集重构模拟信号 $ya(t)$,并将 $x(n)$叠加在它上面。

d. 注意观察重构的结果:在每种情况下有正确的频率但却有不同的振幅。解释这个结果。讨论 $x_a(t)$的相位对信号的采样和重构所起的作用。

第四章 z 变 换

第三章我们学过离散傅利叶变换,这是一种用复指数序列表示离散信号的方法。它通过运用频率响应函数 $H(e^{j\omega})$将系统函数表示在频域中。很明显,这种表示方法用 $H(e^{j\omega})$函数简化了正弦稳态响应的计算,对 LT1 系统很方便。此外,任意给定的绝对收敛序列 $x(n)$的响应都可方便地在频域中计算,并表示成其变换 $X(e^{j\omega})$和频率响应 $H(e^{j\omega})$的乘积形式。但是,傅氏变换有两个缺点,其一,在实际中许多有用的信号,如 $u(n)$和 $nu(n)$...等,它们的离散傅利叶变换不存在。其二,系统对初始条件的暂态响应,或由时变输入引起的系统响应,都无法用离散傅利叶变换方法来计算。

为了克服上述的两个缺点,要把离散傅利叶变换方法进行推广,推广后的方法称为 z 变换。它的单向变换形式可用以获得初始条件和时变输入引起的系统响应,同时它的双向变换形式提供了另一种域,从而可对更多类型的序列和系统进行分析。

双向 z 变换

序列 $x(n)$的 z 变换表示为下式:

$$X(z) \triangleq Z[x(n)] = \sum_{n=-\infty}^{\infty} x(n) z^{-n} \tag{4.1}$$

式中的 z 是复值变量。

使得 $X(z)$存在的所有 z 的值的集合称为收敛域 ROC,表示成:

$$R_{x-} < |z| < R_{x+} \tag{4.2}$$

其中 R_{x-}和 R_{x+}是正整数。

复数函数 $X(z)$的 z 反变换的形式如下式:

$$x(n) \triangleq Z^{-1}[X(z)] = \frac{1}{2\pi j} \oint_C X(z) z^{n-1} dz \tag{4.3}$$

其中 C 是一条逆时针方向并包含于收敛域内的围线。

注解:

1.复数变量 z 称为复数频率,表示为 $z = |z| e^{j\omega}$,其中$|z|$是衰减量,ω 是实际的频率。

2.由于收敛域是由幅度$|z|$定义,所以收敛域的图形如图 4.1 所示,是一个开的环形区

域。其中 R_{x-} 可以取零值，R_{x+} 可取 ∞。

3.如果 $R_{x+} < R_{x-}$，说明收敛域不存在，则 z 变换不存在。

4.若等式 $|z| = 1$（或 $z = e^{j\omega}$）得 z 平面中的一个半径为单位长度的圆，称为单位圆。如果收敛域中包含有单位圆的话，我们就可以在单位圆上求取 $X(z)$ 的值。

$$X(z)|_{z=e^{j\omega}} = X(e^{j\omega}) = \sum_{n=-\infty}^{\infty} x(n)e^{-j\omega} = F[x(n)]$$

因此，可以把 $X(e^{j\omega})$ 的离散付利叶变换看成是 z 变换 $X(z)$ 的特例。

例 4.1 设 $x_1(n) = a^n u(n), 0 < |a| < \infty$，（这种序列称为正时间序列）。然后有

$$X_1(z) = \sum_{0}^{\infty} a^n z^{-n} = \sum_{0}^{\infty} \left(\frac{a}{z}\right)^n = \frac{1}{1 - az^{-1}}; \quad 若\left(\frac{a}{z}\right) < 1$$

$$= \frac{z}{z-a}, \ |z| > |a| \Rightarrow \quad \mathrm{ROC}_1: \underbrace{|a|}_{R_{x-}} < |z| < \underbrace{\infty}_{R_{x+}}$$

说明：本例中的 $X_1(z)$ 是一个有理函数，即，

$$X_1(z) \triangleq \frac{B(z)}{A(z)} = \frac{z}{z-a}$$

其中 $B(z) = z$ 是一个分子多项式，$A(z) = z - a$ 是一个分母多项式。$B(z)$ 的根称为 $X(z)$ 的零点，同样 $A(z)$ 的根称为 $X(z)$ 的极点。此例中，$X_1(z)$ 在原点 $z = 0$ 处有一个零点，在 $z = a$ 处有一个极点。因此，$x_1(n)$ 也可表示成为 z 平面内的零-极点图，如图 4.2 所示，其中'o'表示零点，'x'表示极点。

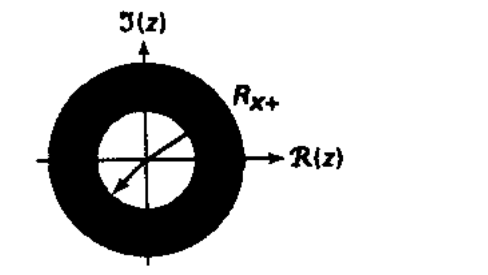


图 4.1 收敛域的一般形式

Im(z)
Re(z)
0

图 4.2 例 4.1 收敛域

例 4.2 设 $x_2(n) = -b^n u(-n-1), 0 < |b| < \infty$，（这种序列称为负时间序列），然后有

$$X_2(z) = -\sum_{-\infty}^{-1} b^n z^{-n} = -\sum_{-\infty}^{-1} \left(\frac{b}{z}\right)^n$$

$$= -\sum_{1}^{\infty} \left(\frac{z}{b}\right)^n = 1 - \sum_{0}^{\infty} \left(\frac{z}{b}\right)^n$$

$$= 1 - \frac{1}{1 - z/b} = \frac{z}{z-b}, \quad \mathrm{ROC}_2: \underbrace{0}_{R_{x-}} < |z| < \underbrace{|b|}_{R_{x+}}$$

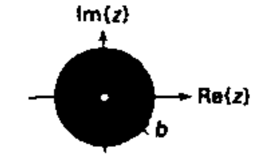


图 4.3 所示即 $x_2(n)$的收敛域 2 及零-极点图。

说明：此例中若 $a=b$，则 $X_2(z)=X_1(z)$，除了它们相应的收敛域不同(即收敛域 1≠收敛域 2)外。这表明收敛域是保证 z 变换单值性的区分特征，因而在系统分析中起着重要的作用。

图 4.3 例 4.2 的收敛域

例 4.3 设 $x_3(n)=x_1(n)+x_2(n)=a^n u(n)-b^n u(-n-1)$(这种序列称为双边序列)，然后利用以上两例：

$$
\begin{aligned}
X_3(z) &= \sum_{n=0}^{\infty} a^n z^{-n} - \sum_{-\infty}^{-1} b^n z^{-n} \\
&= \left\{\frac{z}{z-a}, \mathrm{ROC}_1: |z|>|a|\right\} + \left\{\frac{z}{z-b}, \mathrm{ROC}_1: |z|<|b|\right\} \\
&= \frac{z}{z-a} + \frac{z}{z-b}; \quad \mathrm{ROC}_3: \mathrm{ROC}_1 \cap \mathrm{ROC}_2
\end{aligned}
$$

若$|b|<|a|$，则收敛域 3 是一个空集，$X_3(z)$不存在；若$|a|<|b|$，则收敛域 3 为$|a|<|z|<|b|$，且 $X_3(z)$存在于此区域，如图 4.4 所示。

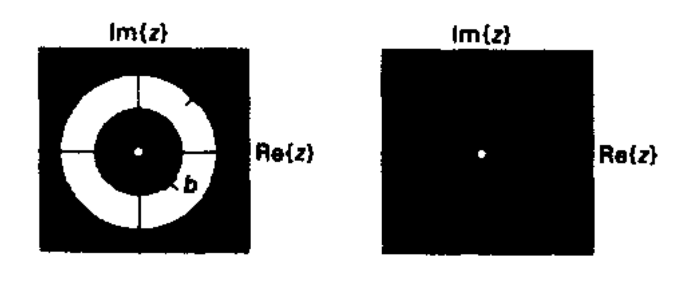


图 4.4 例 4.3 的收敛域

收敛域的性质

通过观察以上三例中的收敛域，我们可以总结出以下性质：

1. 由于收敛条件由$|z|$的幅度决定，所以收敛域总有一个圆的边界。

2. 例 4.1 中的序列 $x_1(n)=a^n u(n)$是右序列的一个特例，当 n 小于某个特定的 n_0 时，$n<n_0$，$x(n)$的值为零。由例 4.1 可以看出，右序列的收敛域总是在半径为 R_{x-} 的圆外。若 $n_0 \geqslant 0$，右序列也称为因果序列。

3. 例 4.2 中的 $x_2(n)=-b^n u(-n-1)$序列是左序列的特例。当 n 大于某个特定的值 n_0 时，$n>n_0$，$x(n)$的值为零。若 $n_0<=0$ 时，所求得的序列称为反因果序列。由例 4.2 可以看出，左序列的收敛域总是在半径为 R_{x+} 的圆内。

4. 例 4.3 中的序列 $x_3(n)$是一个双边序列。若双边序列的收敛域存在，则它必是一个开放的环，其中 $R_{x-}<|z|<R_{x+}$。

5. 如果存在一个序列，它在 $n<n_1$ 和 $n>n_2$ 时取零值，则称为有穷序列。这类序列的收

敛域是整个 z 平面。若 $n_1<0$,则 $z=\infty$ 不属于收敛域;若 $n_2>0$,则 $z=0$ 亦不属于收敛域。

6.由于 $X(z)$ 一致收敛于收敛域,所以其中不可能包含有极点。

7.对于有理函数 $X(z)$,其收敛域边界上至少有一个极点。

8.收敛域是一个连通的区域,即,收敛域不可分割。

在数字信号处理中,由于几乎所有的数字数据都是在实时情况下获得,信号都假设成是由激励引起的。因此,我们唯一关心的收敛域是上面提到的第 2 种。

z 变换的重要特性

z 变换的特性是我们在第三章中学过的离散付里叶变换特性的推广,接下来我们将不加证明地给出 z 变换的若干重要特性。

1.线性特性

$$Z[a_1x_1(n)+a_2x_2(n)]=a_1X_1(z)+a_2X_2(z);\qquad \text{ROC:ROC}_{x1}\cap\text{ROC}_{x2} \tag{4.4}$$

2.样本的移位

$$Z[x(n-n_0)]=z^{-n_0}X(z);\qquad \text{ROC:ROC}_x \tag{4.5}$$

3.频率的移位

$$Z[a^nx(n)]=X\left(\frac{z}{a}\right);\qquad \text{ROC:ROC}_x\text{ 乘 }|a| \tag{4.6}$$

4.折叠

$$Z[x(-n)]=X(1/z);\qquad \text{ROC:ROC}_x\text{ 的倒数} \tag{4.7}$$

5.复共轭序列

$$Z[x^*(n)]=X^*(z^*);\qquad \text{ROC:ROC}_x \tag{4.8}$$

6.z 域中的微分性质

$$Z[nx(n)]=-z\frac{dX(z)}{dz};\qquad \text{ROC:ROC}_x \tag{4.9}$$

7.乘积性质:

$$Z[x_1(n)x_2(n)] = \frac{1}{2\pi j}\oint_C X_1(v)X_2(z/v)v^{-1}dv; \qquad \text{ROC:ROC}_{x1}\cap\text{ROC}_{x2}\text{的倒数} \tag{4.10}$$

8.卷积:

$$Z[x_1(n)*x_2(n)] = X_1(z)X_2(z); \qquad \text{ROC:ROC}_{x1}\cap\text{ROC}_{x2} \tag{4.11}$$

最后一条性质将时间域中的卷积运算变成两个函数的乘积,这个性质在许多方面都很有用。比如 $X_1(z)$和 $X_2(z)$是两个多项式,通过使用 MATLAB 的 conv 函数即可求得它们的积。

例 4.4 有 $X_1(z)=2+3z^{-1}+4z^{-2}$和 $X_2(z)=3+4z^{-1}+5z^{-2}+6z^{-3}$
求 $X_3(z)=X_1(z)X_2(z)$
解:从 z 变换的定义我们看到

$$x_1(n)=\{\underset{\uparrow}{2},3,4\} \quad \text{和} \quad x_2(n)=\{\underset{\uparrow}{3},4,5,6\}$$

从而多项式的积的系数将由上述两个序列的卷积给出。

```
>>x1=[2,3,4];x2=[3,4,5,6];
>>x3=conv(x1,x2)
x3=    6    17    34    43    38    24
```

因此

$$X_3(z)=6+17z^{-1}+34z^{-2}+43z^{-3}+38z^{-4}+24z^{-5}$$

用第二章中开发的 conv_m 函数,同样可以求取非因果序列的两个 z 域多项式的积。
例 4.5 有 $X_1(z)=z+2+3z^{-1}$和 $X_2(z)=2z^2+4z+3+5z^{-1}$。
求 $X_3(z)=X_1(z)X_2(z)$。
解:由于 $x_1(n)=\{1,\underset{\uparrow}{2},3)$和 $x_2(n)=\{2,4,\underset{\uparrow}{3},5)$
用 MATLAB 编程

```
>>x1=[1,2,3];n1=[-1:1];
>>x2=[2,3,4,5];n2=[-2:1];
>>[x3,n3]=conv-m(x1,n1,x2,n2)
x3=
    2    8    17    23    19    15
n3=
```

```
    -3    -2    -1    0    1    2
```

得

$$X_3(z)=2z^3+8z^2+17z+23+19z^{-1}+15z^{-2}$$

前面我们提到用一个多项式去除另一个时,需要一种称为反向卷积的逆运算[19 第六章]。在 MATLAB 中,用语句[p,r] = deconv(b,a)可计算出多项式 b 除以多项式 a,其商为多项式 p,余项为 r。例如,将例 4.4 中的多项式 $X_3(z)$除以 $X_1(z)$。

```
>>x3 = [6,17,34,43,38,24];x1 = [2,3,4];
>>[x2,r] = deconv(x3,x1)
x2 =
    3    4    5    6
r =
    0    0    0    0    0    0
```

从而得到和我们期望相同的多项式 $X_2(z)$的系数。为了获得样本序数,我们需像修正 conv_m 函数一样来修正 deconv 函数。例 4.8 中作了开发。此运算对于从广义有理函数中求取有理部分十分有效。

卷积的第二个重要用途是将其特性应用于系统输出计算中,我们在下一节将看到实例。这种描述对于使用 MATLAB 来验证 z 变换表达式非常有效。我们注意到 MATLAB 是一个数值处理器(除非使用了符号工具箱),所以它不能够直接用于 z 变换的运算。以后我们将详细说明这一点。设序列 $X(n)$有如下有理变换式,

$$X(z)=\frac{B(z)}{A(z)}$$

其中 $B(z)$和 $A(z)$是 z^{-1}形式的多项式。若我们将 $B(z)$和 $A(z)$的系数分别组成序列 b 和序列 a 用于 filter 子程序中,并给此滤波器加上脉冲序列激励 $\delta(n)$,根据式(4.11)和式 $Z[\delta(n)]=1$,滤波器的输出将为 $x(n)$(这是一种计算 z 反变换的数值方法,下一节我们将讨论分析方法)。我们可以将输出与给定的 $x(n)$进行比较,从而确定 $x(n)$的 z 变换是否为 $X(z)$,见例 4.6。

一些常用的 z 变换对

利用 z 变换的定义及其性质,可以得到一些常用序列的 z 变换。表 4.1 列出了一部分这种序列。

例 4.6 用 z 变换性质和 z 变换表求下述序列的 z 变换,

$$x(n)=(n-2)(0.5)^{(n-2)}\cos\left[\frac{\pi}{3}(n-2)\right]u(n-2)$$

表 4.1 常用的变换对

序 列	z 变换	收 敛 域
$\delta(n)$	1	$\forall z$
$u(n)$	$\frac{1}{1-z^{-1}}$	$\lvert z\rvert>1$
$-u(-n-1)$	$\frac{1}{1-z^{-1}}$	$\lvert z\rvert<1$
$a^n n(n)$	$\frac{1}{1-az^{-1}}$	$\lvert z\rvert>\lvert a\rvert$
$-b^n u(-n-1)$	$\frac{1}{1-bz-1}$	$\lvert z\rvert<\lvert b\rvert$
$[a^n\sin\omega_0 n]u(n)$	$\frac{(a\sin\omega_0)z^{-1}}{1-(2a\cos\omega_0)z^{-1}+a^2z^{-2}}$	$\lvert z\rvert>\lvert a\rvert$
$[a^n\cos\omega_0 n]u(n)$	$\frac{1-(a\cos\omega_0)z^{-1}}{1-(2a\cos\omega_0)z^{-1}+a^2z^{-2}}$	$\lvert z\rvert>\lvert a\rvert$
$na^n u(n)$	$\frac{az^{-1}}{(1-az^{-1})^2}$	$\lvert z\rvert>\lvert a\rvert$
$-nb^n u(-n-1)$	$\frac{bz^{-1}}{(1-bz^{-1})^2}$	$\lvert z\rvert<\lvert b\rvert$

解:应用样本移位特性,有

$$X(z)=Z[x(n)]=z^{-2}Z\left[n(0.5)^n\cos\left(\frac{\pi n}{3}\right)u(n)\right]$$

其收敛域不变,利用微分特性,有乘式

$$X(z)=z^{-2}\left\{-z\frac{dZ\left[(0.5)^n\cos\left(\frac{\pi}{3}n\right)u(n)\right]}{dz}\right\}$$

其收敛域不变,现在由表 4.1 查得$(0.5)^n\cos(\frac{\pi}{3}n)u(n)$的 z 变换为

$$Z\left[(0.5)^n\cos\left(\frac{\pi n}{3}\right)u(n)\right]=\frac{1-\left(0.5\cos\frac{\pi}{3}\right)z^{-1}}{1-2\left(0.5\cos\frac{\pi}{3}\right)z^{-1}+0.25z^{-2}};\quad |z|>0.5$$

$$=\frac{1-0.25z^{-1}}{1-0.5z^{-1}+0.25z^{-2}};\quad |z|>0.5$$

因此

$$X(z) = -z^{-1}\frac{d}{dz}\left\{\frac{1-0.25z^{-1}}{1-0.5z^{-1}+0.25z^{-2}}\right\}, \qquad |z|>0.5$$

$$= -z^{-1}\left\{\frac{-0.25z^{-2}+0.5z^{-3}-0.0625z^{-4}}{1-z^{-1}+0.75z^{-2}-0.25z^{-3}+0.0625z^{-4}}\right\}, \qquad |z|>0.5$$

$$= \frac{0.25z^{-3}-0.5z^{-4}+0.0625z^{-5}}{1-z^{-1}+0.75z^{-2}-0.25z^{-3}+0.0625z^{-4}}, \qquad |z|>0.5$$

用 MATLAB 进行验证：通过求取序列 $X(n)$的 8 个样本值，来检验表达式 $X(z)$是否正确。

```
>>b=[0,0,0,0.25,-0.5,0.625];a=[1,-1,0.75,-0.25,0.0625];
>>[delta,n]=impseq(0,0,7)
delta=
    1    0    0    0    0    0    0    0
n =
    0    1    2    3    4    5    6    7
>>x=filter(b,a,delta) % 检验序列
x=
  Columns 1through 4
                  0                  0                  0   0.25000000000000
  Columns 5 through 8
  -0.25000000000000   0.37500000000000   0.12500000000000   0.078125000000000
>>x=[(n-2).*(1/2).^(n-2).*cos(pi*(n-2)/3).*stepseq(2,0,7) % 原始序列
x =
  Columns 1 through 4
                  0                  0                  0   0.25000000000000
  Columns 5 through 8
  -0.25000000000000   0.37500000000000   0.12500000000000   0.07812500000000
```

这个方法可用来进行 z 变换计算的校验。

z 反变换

由定义(4.3)可知，计算 z 反变换需要在一复围线上求积分，通常来讲这是一个复杂的过程。最实际的方法是部分分式分解法，它利用表 4.1 的 z 变换表(或者其他教科书中提出的类似的变换表)进行反变换，但是这就要求 z 变换式必须是一个有理式。在数字信号处理中通常能满足上述要求。

中心思想：若 $X(z)$是 z^{-1}的有理函数，可用部分分式分解的方法将其变成简单因式项(一阶)的和，然后由 z 变换表可写出对应这因式项的序列来。

z 反变换过程总结如下：

方法:给定

$$X(z)=\frac{b_0+b_1z^{-1}+\ldots+b_Mz^{-M}}{1+a_1z^{-1}+\ldots+a_Nz^{-N}},R_{x-}<|z|<R_{x+} \tag{4.12}$$

• 将之化成

$$X(z)=\underbrace{\frac{\tilde{b}_0+\tilde{b}_1z^{-1}+\ldots+\tilde{b}_{N-1}z^{-(N-1)}}{1+a_1z^{-1}+\ldots+a_Nz^{-N}}}_{\text{真有理项}}+\underbrace{\sum_{k=0}^{M-N}C_kz^{-k}}_{\text{若}M\geq N\text{多项式}}$$

式子右边第一项是真有理式部分,第二项是多项式(无穷项)部分。若 $M\geqslant N$,用 deconv 函数可实现上述多项式的除法。

• 对 $X(z)$真有理式部分进行部分分式展开,得

$$X(z)=\sum_{k=1}^{N}\frac{R_k}{1-p_kz^{-1}}+\underbrace{\sum_{k=0}^{M-N}C_kz^{-k}}_{M\geq N} \tag{4.13}$$

此处 p_k 是 $X(z)$的第 k 个极点,R_k 是在 p_k 处的留数。假设都是单极点,留数由下式给出:

$$R_k=\frac{\tilde{b}_0+\tilde{b}_1z^{-1}+\ldots+\tilde{b}_{N-1}z^{-(N-1)}}{1+a_1z^{-1}+\ldots+a_Nz^{-N}}(1-p_kz^{-1})\Bigg|_{z=p_k}$$

多重极点的展开有更通用的形式。若 p_k 是一个 r 重的极点,则其分解式如下:

$$\sum_{l=1}^{r}\frac{R_{k,l}z^{-(l-1)}}{(1-p_kz^{-1})^l}=\frac{R_{k,1}}{1-p_kz^{-1}}+\frac{R_{k,2}z^{-1}}{(1-p_kz^{-1})^2}+\ldots+\frac{R_{k,r}z^{-(r-1)}}{(1-p_kz^{-1})^r} \tag{4.14}$$

其中留数 $R_{k,l}$的计算可使用[19]中提到的更通用的公式。

• 记 $X(n)$为

$$X(n)=\sum_{k=1}^{N}R_kZ^{-1}\left[\frac{1}{1-p_kz^{-1}}\right]+\underbrace{\sum_{k=0}^{M-N}C_k\delta(n-k)}_{M\geq N}$$

• 用表 4.1 给出的关系最终求得 $x(n)$。

$$Z^{-1}\left[\frac{z}{z-p_k}\right]=\begin{cases} p_k^n u(n) & |z_k| \le R_{x-} \\ -p_k^n u(-n-1) & |z_k| \ge R_{x+} \end{cases} \tag{4.15}$$

例 4.7 求取 $x(z)=\dfrac{z}{3z^2-4z+1}$ 的 z 变换。

解 记

$$X(z)=\frac{z}{3(z^2-\frac{4}{3}z+\frac{1}{3})}=\frac{\frac{1}{3}z^{-1}}{1-\frac{4}{3}z^{-1}+\frac{1}{3}z^{-2}}$$

$$=\frac{\frac{1}{3}z^{-1}}{(1-z^{-1})(1-\frac{1}{3}z^{-1})}=\frac{\frac{1}{2}}{1-z^{-1}}-\frac{\frac{1}{2}}{1-\frac{1}{3}z^{-1}}$$

或

$$X(z)=\frac{1}{2}\left(\frac{1}{1-z^{-1}}\right)-\frac{1}{2}\left(\frac{1}{1-\frac{1}{3}z^{-1}}\right)$$

由此知，$X(z)$有两个极点：$z_1=1$ 及 $z_2=1/3$；由于收敛域未给定，所以图 4.5 中绘出了三种可能的收敛域。

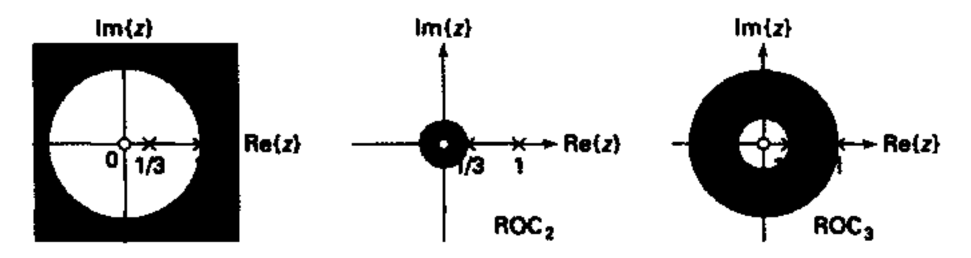


图 4.5 例 4.7 的收敛域

a. 收敛域 1 (ROC1)： $1<|z|<\infty$，此处的两个极点都在收敛域 1(ROC_1)的内部，所以有 $|z_1|<R_{x-}=1$ 和 $|z_2|\leqslant 1$，因此由(4.15)式

$$x_1(n)=\frac{1}{2}u(n)-\frac{1}{2}\left(\frac{1}{3}\right)^n u(n)$$

知其是一个右序列。

b. 收敛域 2(ROC_2)：$0<|z_1|<1/3$，此处的两个极点都在收敛域 2(ROC_2)的外部，所以有 $|z_1|>=R_{x+}=1/3$ 和 $|z_2|>=1/3$，因此由(4.15)式

$$x_2(n)=\frac{1}{2}\{-u(-n-1)\}-\frac{1}{2}\left\{-\left(\frac{1}{3}\right)^n u(-n-1)\right\}$$
$$=\frac{1}{2}\left(\frac{1}{3}\right)^n u(-n-1)-\frac{1}{2}u(-n-1)$$

知其是一个左序列。

c. 收敛域 3(ROC_3):$1/3<|z|<1$,此处的极点 z_1 都在收敛域 3 (ROC_3)的外部,即 $|z_1|\geqslant R_{x+}=1$;而极点 z_2 在收敛域 3 的内部,即 $|z_2|\leqslant 1/3$。因此由(4.15)式

$$x_3(n)=-\frac{1}{2}u(-n-1)-\frac{1}{2}\left(\frac{1}{3}\right)^n u(n)$$

知其是一个双边序列。

MATLAB 实现

MATLAB 中的 residuez 函数可计算出有理函数的留数部分和直接(或多项式)项。设有多项式如下:

$$X(z)=\frac{b_0+b_1z^{-1}+\dots+b_Mz^{-M}}{a_0+a_1z^{-1}+\dots+a_Nz^{-N}}=\frac{B(z)}{A(z)}=\sum_{k=1}^{N}\frac{R_k}{1-p_kz^{-1}}+\underbrace{\sum_{k=0}^{M-N}C_kz^{-k}}_{M\geq N}$$

其分子分母都按 z^{-1}的递增顺序排列。用语句[R,p,c] = residuez(b,a)可求得 $X(z)$的留数、极点和直接项,分子分母多项式 $A(z)$,$B(z)$分别由矢量 a,b 给定。求得的列向量 R 包含着留数;列向量 p 包含着极点的位置;行向量 C 包含着直接项。如果 p(k) = ... = p(k + r - 1)是一 r 重极点,则其展开形式包括如下形式的项:

$$\frac{R_k}{1-p_kz^{-1}}+\frac{R_{k+1}}{(1-p_kz^{-1})^2}+\dots+\frac{R_{k+r-1}}{(1-p_kz^{-1})^r} \tag{4.16}$$

这种形式与式(4.14)不同。

类似的,函数[b,a] = residuez(R,p,c),有三个输入变量和两个输出变量,它把部分分式变换成多项式的系数行向量 b 和 a。

例 4.8 为了验证留数函数,我们来看例 4.7 中的有理方程。

$$X(z)=\frac{z}{3z^2-4z+1}$$

解 首先将 $X(z)$按 z^{-1}的升幂排列,

$$X(z)=\frac{z^{-1}}{3-4z^{-1}+z^{-2}}=\frac{0+z^{-1}}{3-4z^{-1}+z^{-2}}$$

用 MATLAB。

```
>> b=[0,1]; a=[3,-4,1];
>> [R,p,C]=residuez(b,a)
R =
    0.5000
   -0.5000
p =
    1.0000
    0.3333
C =
    []
```

如前,我们得到

$$X(z)=\frac{\frac{1}{2}}{1-z^{-1}}-\frac{\frac{1}{2}}{1-\frac{1}{3}z^{-1}}$$

类似的,将其变成有理方程。

```
>>[b a]=residuez(R,p,C)
b =
    0.0000
    0.3333
a =
    1.0000
   -1.3333
    0.3333
```

由此得到原来的形式

$$X(z)=\frac{0+\frac{1}{3}z^{-1}}{1-\frac{4}{3}z^{-1}+\frac{1}{3}z^{-2}}=\frac{z^{-1}}{3-4z^{-1}+z^{-2}}=\frac{z}{3z^{2}-4z+1}$$

例 4.9 计算下式的反变换

$$X(z)=\frac{1}{(1-0.9z^{-1})^2(1+0.9z^{-1})},\qquad |z|>0.9$$

解 用MATLAB可求出分母多项式的极点和其留数。

```
>>b=1; a=poly([0.9,0.9,-0.9])
a =
    1.0000    -0.9000    -0.8100    0.7290
>>[R,p,C]=residuez(b,a)
R =
    0.2500
    0.5000
    0.2500
p =
    0.9000
    0.9000
   -0.9000
C =
    []
```

注意到分母计算用到MATLAB的多项式函数poly,它可以在给出其根时求出多项式的系数,也可用conv函数,但此处用poly函数更方便。通过留数计算和式(4.16)中留数的阶次,得到

$$\begin{aligned}X(z)&=\frac{0.25}{1-0.9z^{-1}}+\frac{0.5}{(1-0.9z^{-1})^2}+\frac{0.25}{1+0.9z^{-1}},\qquad |z|>0.9\\&=\frac{0.25}{1-0.9z^{-1}}+\frac{0.5}{0.9}z\frac{(0.9z^{-1})}{(1-0.9z^{-1})^2}+\frac{0.25}{1+0.9z^{-1}},\qquad |z|>0.9\end{aligned}$$

因此由表4.1,并用z变换的时域移位性质,得

$$x(n)=0.25(0.9)^nu(n)+\frac{5}{9}(n+1)(0.9)^{n+1}u(n+1)+0.25(-0.9)^nu(n)$$

简化上式得

$$x(n)=0.75(0.9)^nu(n)+0.5n(0.9)^nu(n)+0.25(-0.9)^nu(n)$$

MATLAB验证

```
>> [delta,n] = impseq(0,0,7);
>> x = filter(b,a,delta) % 检验序列
x =
  Columns 1 through 4
  1.00000000000000  0.90000000000000  1.62000000000000  1.45800000000000
  Columns 5 through 8
  1.96830000000000  1.77147000000000  2.12576400000000  1.91318760000000
>> x = 0.75 * (0.9).^n + 0.5 * n. * (0.9).^n + 0.25 * ( - 0.9).^n % 原始序列
x =
  Columns 1 through 4
  1.00000000000000  0.90000000000000  1.62000000000000  1.45800000000000
Columns 5 through 8
  1.96830000000000  1.77147000000000  2.12576400000000  1.91318760000000
```

例 4.10 求下式的 z 的反变换

$$X(z)=\frac{1+0.4\sqrt{2}z^{-1}}{1-0.8\sqrt{2}z^{-1}+0.64z^{-2}}$$

并使求出的序列是因果序列，且不含有复数。

解 为确定因果序列的收敛域，首先必须求出极坐标形式下的 $X(z)$的极点。

```
>> b = [1,0.4 * sqrt(2)]; a = [1, - 0.8 * sqrt(2),0.64];
>> [R,p,C] = residuez(b,a)
R =
    0.5000 - 1.0000i
    0.5000 + 1.0000i
p =
    0.5657 + 0.5657i
    0.5657 - 0.5657i
C =
    []
>> Mp = abs(p')      % 极点幅值
Mp =
    0.8000      0.8000
>> Ap = angle(p')/pi     % 极点幅角(以 pi 为单位)
Ap =
   - 0.2500      0.2500
```

从上面的计算得出

$$X(z)=\frac{0.5+j}{1-|0.8|e^{-j\frac{\pi}{4}}z^{-1}}+\frac{0.5-j}{1-|0.8|e^{j\frac{\pi}{4}}z^{-1}},\quad |z|>0.8$$

并由表 4.1,得到

$$\begin{aligned}x(n)&=(0.5+j)|0.8|^n e^{-j\frac{\pi}{4}n}u(n)+(0.5-j)|0.8|^n e^{j\frac{\pi}{4}n}u(n)\\&=|0.8|^n[0.5\{e^{-j\frac{\pi}{4}n}+e^{j\frac{\pi}{4}n}\}+j\{e^{-j\frac{\pi}{4}n}+e^{j\frac{\pi}{4}n}\}]u(n)\\&=|0.8|^n\left[\cos\left(\frac{\pi n}{4}\right)+2\sin\left(\frac{\pi n}{4}\right)\right]u(n)\end{aligned}$$

MATLAB 验证

```
>> [delat, n]=impseq(0,0,6);
>> x=filter(b,a,delta) % 检验序列
x =
  Columns 1 through 4
    1.00000000000000   1.69705627484771   1.28000000000000   0.36203867196751
  Columns 5 through 8
  -0.40960000000000  -0.69511425017762  -0.52428800000000  -0.14829104003789
>> x=((0.8).^n).*(cos(pi*n/4)+2*sin(pi*n/4))
  x=
  Columns 1 through 4
    1.00000000000000   1.69705627484771   1.28000000000000   0.36203867196751
  Columns 5 through 8
  -0.40960000000000  -0.69511425017762  -0.52428800000000  -0.14829104003789
```

z 域中的系统描述

与频响函数 $H(e^{j\omega})$处的情况类似,可以定义 z 域函数 $H(z)$,称为系统函数。但是,与 $H(e^{j\omega})$不同的是,对于不一定 BIBO 稳定的系统,系统函数 $H(z)$仍然存在。

定义 1　系统函数

系统函数 $H(z)$由下式给出

$$H(z)\triangleq Z[h(n)]=\sum_{-\infty}^{\infty}h(n)z^{-n};\quad R_{h-}<|z|<R_{h+}\tag{4.17}$$

用 z 变换的卷积性质式(4.11),输出变换 $Y(z)$由下式给定

$$Y(z) = H(z)X(z) \quad : \mathrm{ROC}_y = \mathrm{ROC}_h \cap \mathrm{ROC}_x \tag{4.18}$$

假定收敛域 x 与收敛域 h 重叠。因此一个线性时不变系统可在 z 域中表示成下面的形式,

$$X(z) \longrightarrow \boxed{H(z)} \longrightarrow Y(z) = H(z)X(z)$$

差分方程描述的系统函数

LT1 系统若用差分方程描述时

$$y(n) + \sum_{k=1}^{N} a_k y(n-k) = \sum_{l=0}^{M} b_l x(n-l) \tag{4.19}$$

其系统函数 $H(z)$很容易就可算出。对方程的两边进行 z 变换,并利用 z 变换的性质,有

$$Y(z) = \sum_{k=1}^{N} a_k z^{-k} Y(z) = \sum_{l=0}^{M} b_l z^{-l} X(z)$$

或

$$H(z) \triangleq \frac{Y(z)}{X(z)} = \frac{\sum\limits_{l=0}^{M} b_l z^{-l}}{1 + \sum\limits_{k=1}^{N} a_k z^{-k}} = \frac{B(z)}{A(z)} \tag{4.20}$$

$$= \frac{b_0 z^{-M}\left(z^M + \ldots + \frac{b_M}{b_0}\right)}{z^{-N}(z^N + \ldots + a_N)}$$

分解因式后得

$$H(z) = b_0 z^{N-M} \frac{\prod\limits_{l=1}^{N}(z - z_l)}{\prod\limits_{k=1}^{N}(z - p_k)} \tag{4.21}$$

此处 z_l 是系统的零点,p_k 是系统的极点。因此,$H(z)$函数(因而一个 LT1 系统)可在 z 域中用零极点图的形式来描述。这个事实在设计简单的滤波器时很重要,只要正确地配置零极点就可达到目的。

分别对有理函数 $H(z)$的分子分母多项式使用 MATLAB 的 roots 函数,就可求得其零极点。(前面我们讨论过 roots 的逆向函数 poly,用它可通过其根求得多项式的系数)。也可用 DSP 工具箱中的 pzplotz(b,a)函数,由给定的分子行向量 b 和分母行向量 a 绘制成系统的零极

点图。同前面一样，符号“o”表示零点，“x”表示极点。图中还给出了用作参考的单位圆。**

传递函数描述

若函数 $H(z)$的收敛域包括单位圆($z=e^{j\omega}$)，我们就可在这个单位圆上计算 $H(z)$，并得到一个频响函数或传递函数 $H(e^{j\omega})$。由式 4.21 得

$$H(e^{j\omega}) = b_0 e^{j(N-M)\omega} \frac{\prod_{1}^{M}(e^{j\omega}-z_l)}{\prod_{1}^{N}(e^{j\omega}-p_k)} \tag{4.22}$$

因子$(e^{j\omega}-z_l)$可解释为成为 z 复平面中由零点 z_l 指向单位圆上 $z=e^{j\omega}$处的向量，同样因子$(e^{j\omega}-p_k)$可表示成为 z 复平面中由极点 p_k 指向单位圆上 $z=e^{j\omega}$处的向量，如图 4.6 所示。因此幅值响应函数如下式

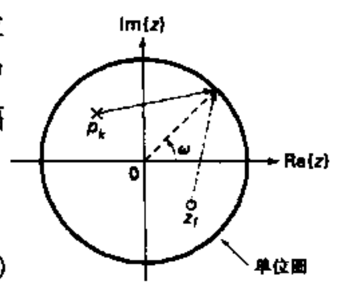


图 4.6　极点和零点向量

$$|H(e^{j\omega})| = |b_0| \frac{|e^{j\omega}-z_1|\cdots|e^{j\omega}-z_M|}{|e^{j\omega}-p_1|\cdots|e^{j\omega}-p_N|} \tag{4.23}$$

上式可看成所有零点到单位圆矢量长度的积除以所有极点到单位圆矢量长度的积，再乘以$|b_0|$。类似地，相角响应函数可以看成是三个部分的线性和，分别是：常数项、线性相位项与非线性相位项（即所有“零点到单位圆的矢量”的相角和与所有“极点到单位圆的矢量”的相角和的差）。

$$\angle H(e^{j\omega}) = \underbrace{[0 \text{ or } \pi]}_{\text{常数项}} + \underbrace{[(N-M)\omega]}_{\text{线性项}} + \underbrace{\sum_{1}^{M}\angle(e^{j\omega}-z_k) - \sum_{1}^{N}\angle(e^{j\omega}-p_k)}_{\text{非线性项}} \tag{4.24}$$

MATLAB 实现

在第三章中，通过直接执行函数的形式，我们已用 MATLAB 画过幅度和相位响应。MATLAB 还提供了一个等效的函数 freqz，基于上面的介绍。该函数的最简调用形式为

```
[H,w] = freqz(b,a,N)
```

b 和 a 分别表示分子和分母的系数向量，此函数返回该系统的 N 点频率矢量 w 和 N 点复数频率响应矢量 H。在上半个单位圆的等间距的 N 个点上计算频率响应。注意矢量 b 和 a 与 filter 函数中用的相同，也可以从差分方程表达式(4.19)导出。第二种形式

** 译者注：原书中采用 zplane 函数，它是新版(2.01 以上)信号处理工具箱中的函数，考虑到读者使用的方便，我们改用 DSP 工具箱中原有的函数 pzplotz(b,a)

```
[H,W] = freqz(b,a,N,'whole')
```

用整个单位圆上等间距的 N 个点计算。还有另一种形式

```
H = freqz(b,a,w)
```

它返回矢量 w 指定的那些频率点上的频率响应,通常在 0 到 π 之间。

例 4.11 已知因果系统

$$y(n)=0.9y(n-1)+x(n)$$

a. 求 $H(z)$并画出零-极点示意图。
b. 画出 $|H(e^{j\omega})|$和$\angle H(e^{j\omega})$。
c. 求脉冲响应 $h(n)$。

解:差分方程可以变形为

$$y(n)-0.9y(n-1)=x(n)$$

a. 根据(4.21)式

$$H(z)=\frac{1}{1-0.9z^{-1}};\ |a|>0.9$$

由于系统是因果的。有一个位于 0.9 的极点和一个位于原点的零点。下面用 MATLAB 举例说明 pzplotz 函数的用法。

```
>> b=[1,0];a=[1,-0.9];
>> pzplotz(b,a)
```

注意我们令 $b=[1,0]$,而不是 $b=1$,是因为 pzplotz 函数假设标量是零点或极点。所作零极点图示于图 4.7 中。

b. 利用(4.23)和(4.24)式,可以确定 $|H(e^{j\omega})|$ 的幅度和相位。下面将用 MATLAB 举例说明 freqz 函数的用法。使用第一种形式,沿单位圆的上半圆取 100 个点。

```
>>[H,w] = freqz(b,a,100);
>>magH = abs(H); phaH = angle(H);
>>subplot(2,1,1); plot(w/pi,magH); grid
>>xlabel('频率 单位:pi'); ylabel('幅度');
>>title('幅度响应');
```

```
>>subplot(2,1,2); plot(w/pi,phaH/pi); grid
>>xlabel('频率 单位:pi'); ylabel('相位 单位:pi');
>>title('相位响应')
```

响应示于图 4.8 中。如果你仔细研究此图，将会发现计算区间是 $0 \leqslant \omega \leqslant 0.99\pi$ 而在 $\omega = \pi$ 点不正确。这是由于在 MATLAB 中，单位圆的下半圆是从 $\omega = \pi$ 开始的。为了克服这个问题，我们使用 freqz 函数的第二种形式，如下：

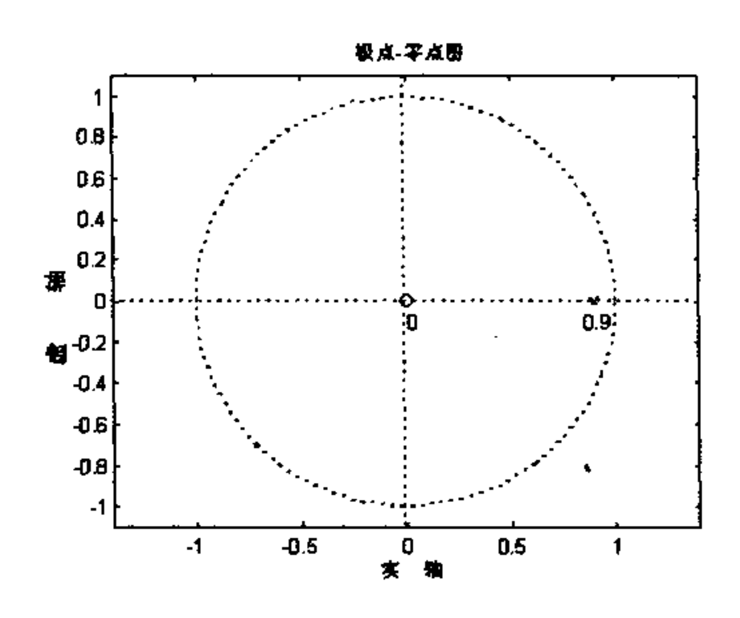


图 4.7 例 4.11a 的零极点图

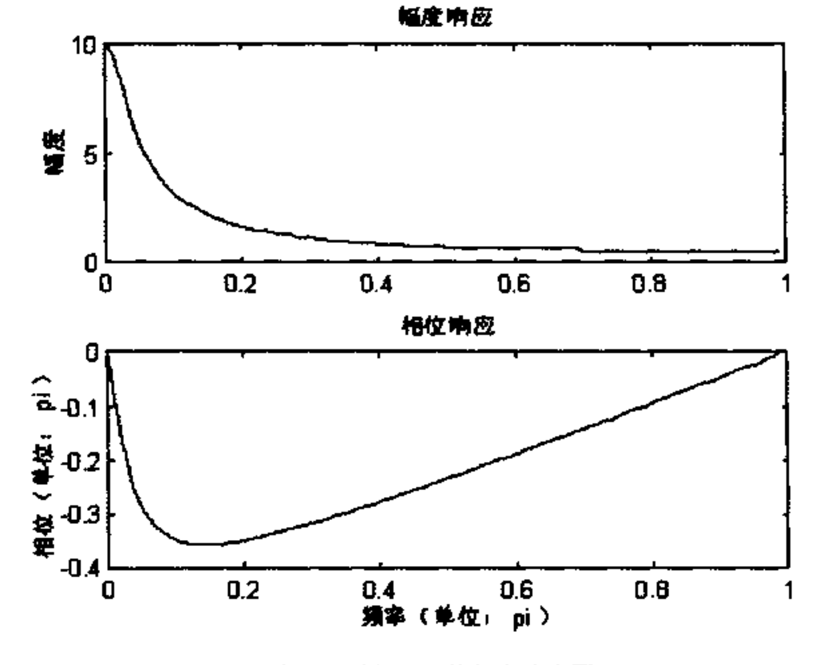


图 4.8 例 4.11 的频率响应图

```
>>[H,w]=freqz(b,a,200,'whole');
```

```
>> magH = abs(H(1:101));phaH = angle(H(1:101));
```

数组 H 的第 101 个元素对应于 $\omega=\pi$。使用 freqz 的第三种形式可以获得类似的结果。

```
>> w = [0:1:100] * pi/100;
>> H = freqz(b,a,W);
>> magH = abs(H); phaH = angle(H);
```

可以根据自己的习惯,使用这三种形式的任何一种。还应注意,这里作图时把 w 和 phaH 数组除以 π,这样坐标轴以 π 为单位,易于读数。**我们强烈推荐这种作法。**

c. 根据 z-变换表 4.1

$$h(n)=Z^{-1}\left[\frac{1}{1-0.9z^{-1}},|z|>0.9\right]=(0.9)^n u(n)$$

例 4.12 已知

$$H(z)=\frac{z+1}{z^2-0.9z+0.81}$$

是因果系统,求

a. 传递函数表达式。

b. 差分方程表达式。

c. 脉冲响应表达式。

解: 系统函数的极点是在 $z=0.9\angle\pm\pi/3$ 处。因此这个因果系统的收敛域是 $|z|>0.9$。所以单位圆在收敛域内,其离散傅里叶变换 $H(e^{j\omega})$存在。

a. 把 $z=e^{j\omega}$代入 $H(z)$。

$$H(e^{j\omega})=\frac{e^{j\omega}+1}{e^{j2\omega}-0.9e^{j\omega}+0.81}=\frac{e^{j\omega}+1}{(e^{j\omega}-0.9e^{j\pi/3})(e^{j\omega}-0.9e^{-j\pi/3})}$$

b. 利用 $H(z)=Y(z)/X(z)$

$$\frac{Y(z)}{X(z)}=\frac{z+1}{z^2-0.9z+0.81}\left(\frac{z^{-2}}{z^{-2}}\right)=\frac{z^{-1}+z^{-2}}{1-0.9z^{-1}+0.81z^{-2}}$$

交叉相乘:

$$Y(z)-0.9z^{-1}Y(z)+0.81z^{-2}Y(z)=z^{-1}X(z)+z^{-2}X(z)$$

作 z 反变换:

$$y(n)-0.9y(n-1)+0.81y(n-2)=x(n-1)+x(n-2)$$

或

$$y(n)=0.9y(n-1)-0.81y(n-2)+x(n-1)+x(n-2)$$

c. 利用 MATLAB。

```
>>b=[0,1,1]; a=[1,-0.9,0.81];
>>[R,p,C]=residuez(b,a)

R =
    -0.6173 - 0.9979i
    -0.6173 + 0.9979i
p =
   0.4500 + 0.7794i
   0.4500 - 0.7794i
C =
    1.2346
>>Mp=abs(p')
Mp =
    0.9000   0.9000
>>Ap=angle(p')/pi
Ap =
    -0.3333   0.3333
```

得到

$$H(z)=1.2346+\frac{-0.6173+j0.9979}{1-|0.9|e^{-j\pi/3}z^{-1}}+\frac{-0.6173-j0.9979}{1-|0.9|e^{j\pi/3}z^{-1}},|z|>0.9$$

因此根据表 4.1,有

$$\begin{aligned}h(n)&=1.2346\delta(n)+[(-0.6173+j0.9979)|0.9|^{n}e^{-j\pi n/3}\\&\quad+(-0.6173-j0.9979)|0.9|^{n}e^{j\pi n/3}]u(n)\\&=1.2346\delta(n)+|0.9|^{n}[-1.2346\cos(\pi n/3)+1.9958\sin(\pi n/3)]u(n)\\&=|0.9|^{n}[-1.2346\cos(\pi n/3)+1.9958\sin(\pi n/3)]u(n-1)\end{aligned}$$

最后一步是由于 $h(0)=0$。

各种系统表达式之间的关系

在本章和前两章中，我们阐述了几种系统表达式。图 4.9 描述了它们之间的关系。

稳定性和因果性

对于 LTI 系统，BIBO 稳定性等效于 $\sum_{-\infty}^{\infty}|h(k)|<\infty$.。由于离散傅里叶变换存在，这种稳定性就意味着 $H(e^{j\omega})$存在，也就意味着单位圆$|z|=1$在 $H(z)$的收敛域内。这就叫做 z 域稳定性定理；因此只有当系统稳定时，图 4.9 中的虚线才存在。

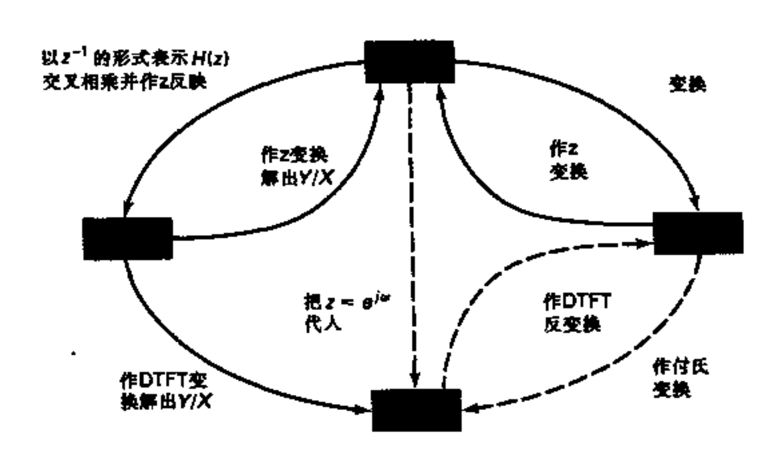


图 4.9 系统表达式的图示

定理 2 z 域 LTI 稳定性

一个 LTI 系统是稳定的，当且仅当单位圆在 $H(z)$的收敛域内时。

LTI 因果性要求：当 $n<0$ 时，$h(n)=0$(就是说，是一个右序列)。这就意味着 $H(z)$的收敛域必须是半径为 R_{h-} 的圆外。这不是充分条件，由于任何右序列都有类似的收敛域。然而当系统稳定时，其因果性易于判断。

定理 3 z 域因果 LTI 稳定性

一个因果 LTI 系统是稳定的，当且仅当系统函数 $H(z)$的所有极点都在单位圆内时。

例 4.13 已知一个由差分方程描述的因果 LTI 系统：

$$y(n)=0.81y(n-2)+x(n)-x(n-2)$$

求：**a.** 系统函数 $H(z)$，

b. 单位脉冲响应 $h(n)$，

c. 单位阶跃响应 $v(n)$，也就是对单位阶跃 $u(n)$的响应，

d. 频率响应函数 $H(e^{j\omega})$，并画出在 $0\leqslant\omega\leqslant\pi$ 区间的幅频和相频曲线。

解：由于系统是因果的，收敛域是半径等于极点最大幅度的圆外。

a.对差分方程两边同时作 z 变换，解出 $Y(z)/X(z)$，或利用(4.20)，得到

$$H(z)=\frac{1-z^{-2}}{1-0.81z^{-2}}=\frac{1-z^{-2}}{(1+0.9z^{-1})(1-0.9z^{-1})},\quad |z|>0.9$$

b.利用 MATLAB 作部分分式展开

```
>> b=[1,0,-1]; a=[1,0,-0.81];
>> [R,p,C]=residuez(b,a)
R =
  -0.1173
  -0.1173
p =
   0.9000
  -0.9000
C =
   1.2346
```

得到

$$H(z)=1.2346-0.1173\frac{1}{1+0.9z^{-1}}-0.1173\frac{1}{1-0.9z^{-1}},|z|>0.9$$

或根据表 4.1

$$h(n)=1.2346\delta(n)-0.1173\{1+(-1)^n\}(0.9)^n u(n)$$

c.根据表 4.1 $\quad Z[u(n)]=\frac{1}{1-z^{-1}},|z|>1$.因此

$$V(z)=H(z)U(z)=\left[\frac{(1+z^{-1})(1-z^{-1})}{(1+0.9z^{-1})(1-0.9z^{-1})}\right]\left[\frac{1}{1-z^{-1}}\right],\ |z|>0.9\cap|z|>1$$

$$=\frac{1+z^{-1}}{(1+0.9z^{-1})(1-0.9z^{-1})},\qquad |z|>0.9$$

或

$$V(z)=1.0556\frac{1}{1-0.9z^{-1}}-0.0556\frac{1}{1+0.9z^{-1}},\ |z|>0.9$$

最后，

$$v(n)=[1.0556(0.9)^n-0.0556(-0.9)^n]u(n)$$

注意，在 $V(z)$的计算过程中存在 $z=1$ 处的零极点对消。这有两种含义。第一，$V(z)$的收敛域仍然是$\{|z|>0.9\}$，而不是$\{|z|>0.9\cap|z|>1=|z|>1\}$。第二，阶跃响应 $v(n)$中没包含稳定项 $u(n)$。

d. 把 $z=e^{j\omega}$代入 $H(z)$

$$H(e^{j\omega})=\frac{1-e^{-j2\omega}}{1-0.81e^{-j2\omega}}$$

下面用 MATLAB 计算并作图

```
> > w = [0:1:500] * pi/500;
> > H = freqz(b,a,w);
> > magH = abs(H); phaH = angle(H);
> > subplot(2,1,1); plot(w/pi,magH); grid
> > xlabel('频率 单位:pi'); ylabel('幅度')
> > title('幅度响应');
> > subplot(2,1,2); plot(w/pi,phaH/pi); grid
> > xlabel('频率 单位:为 pi'); ylabel('相位 单位为:pi')
> > title('相位响应')
```

频率响应曲线示于图 4.10 中。

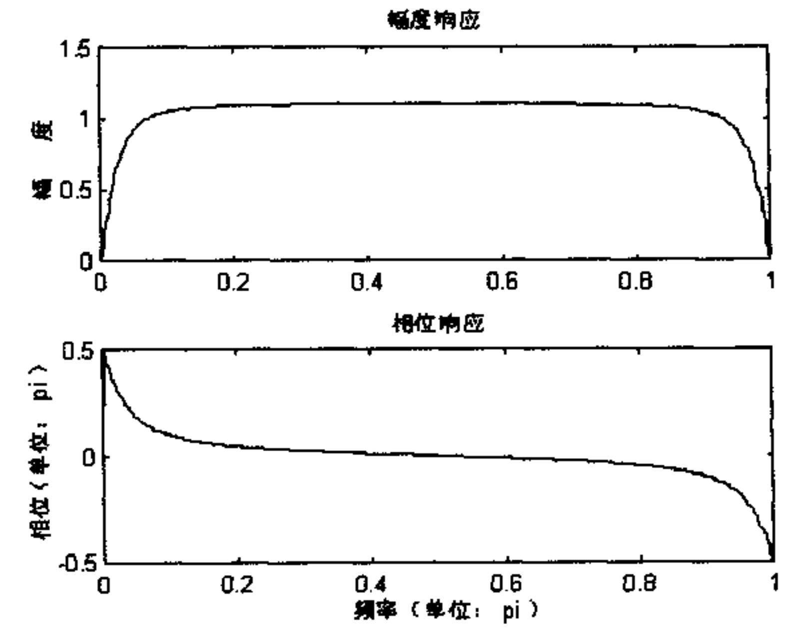


图 4.10　例 4.13 的频率响应曲线

差分方程的解

在第二章中,我们提到了线性常系数差分方程解的两种形式。一种形式包括通解和特解,另一种形式包括零输入(初始状态)响应和零状态响应。使用 z 变换,又提供了一种得到这两种形式的方法。另外,我们也将讨论暂态响应和稳态响应。在数字信号处理中,差分方程通常按正 n 方向给出。因此,解的时间范围是 $n \geqslant 0$。为此,我们定义双边 z 变换的一个变形,叫做单边 z 变换。

定义 4　单边 z 变换

序列 $x(n)$ 的单边 z 变换表示为

$$Z^{+}[x(n)] \triangleq Z[x(n)u(n)] \triangleq X^{+}[z] = \sum_{n=0}^{\infty} x(n)z^{-n} \tag{4.25}$$

其时移特性如下

$$\begin{aligned} Z^{+}[x(n-k)] &= Z[x(n-k)u(n)] \\ &= \sum_{n=0}^{\infty} x(n-k)z^{-n} = \sum_{m=-k}^{\infty} x(m)z^{-(m+k)} \\ &= \sum_{m=-k}^{-1} x(m)z^{-(m+k)} + \left[\sum_{m=0}^{\infty} x(m)z^{-m}\right]z^{-k} \end{aligned}$$

或

$$Z^{+}[x(n-k)] = x(-1)z^{1-k} + x(-2)z^{2-k} + \cdots + x(-k) + z^{-k}X^{+}(z) \tag{4.26}$$

上面的结果可用于求解具有非零初始条件或输入可变的差分方程。我们要解如下差分方程

$$1 + \sum_{k=1}^{N} a_k y(n-k) = \sum_{m=0}^{M} b_m x(n-m), \qquad n \geq 0$$

初始条件为

$$\{y(i), i=-1,\ldots,-N\} \text{ 和 } \{x(i), i=-1,\ldots,-M\}。$$

下面用例题说明解法。

例 4.14　求解

$$y(n)-\frac{3}{2}y(n-1)+\frac{1}{2}(n-2)=x(n),n\geq 0$$

其中

$$x(n)=\left(\frac{1}{4}\right)^n u(n)$$

初始条件为 $y(-1)=4$ 和 $y(-2)=10$。

解:对差分方程的两边同时进行单边 z 变换,得到

$$Y^+(z)-\frac{3}{2}[y(-1)+z^{-1}Y^+(z)]+\frac{1}{2}[y(-2)+z^{-1}y(-1)+z^{-2}Y^+(z)]=\frac{1}{1-\frac{1}{4}z^{-1}}$$

代入初始条件并整理,得

$$Y^+(z)[1-\frac{3}{2}z^{-1}+\frac{1}{2}z^{-2}]=\frac{1}{1-\frac{1}{4}z^{-1}}+(1-2z^{-1})$$

或

$$Y^+(z)=\frac{\frac{1}{1-\frac{1}{4}z^{-1}}}{1-\frac{3}{2}z^{-1}+\frac{1}{2}z^{-2}}+\frac{1-2z^{-1}}{1-\frac{3}{2}z^{-1}+\frac{1}{2}z^{-2}} \tag{4.27}$$

最后,

$$Y^+(z)=\frac{2-\frac{9}{4}z^{-1}+\frac{1}{2}z^{-2}}{\left(1-\frac{1}{2}z^{-1}\right)(1-z^{-1})\left(1-\frac{1}{4}z^{-1}\right)}$$

进行部分分式展开,得到

$$Y^+(z)=\frac{1}{1-\frac{1}{2}z^{-1}}+\frac{\frac{2}{3}}{1-z^{-1}}+\frac{\frac{1}{3}}{1-\frac{1}{4}z^{-1}} \tag{4.28}$$

反变换后，得到解为

$$y(n)=\left[\left(\frac{1}{2}\right)^n+\frac{2}{3}+\frac{1}{3}\left(\frac{1}{4}\right)^n\right]u(n) \tag{4.29}$$

解的形式 上面的解是差分方程的全响应。可以表示为几种形式。

- 通解和特解：

$$y(n)=\underbrace{\left[\left(\frac{1}{2}\right)^n+\frac{2}{3}\right]u(n)}_{\text{通解}}+\underbrace{\frac{1}{3}\left(\frac{1}{4}\right)^n u(n)}_{\text{特解}}$$

通解由系统极点确定，特解由输入信号的极点确定。

- 暂态响应和稳态响应：

$$y(n)=\underbrace{\left[\frac{1}{3}\left(\frac{1}{4}\right)^n+\left(\frac{1}{2}\right)^n\right]u(n)}_{\text{暂态响应}}+\underbrace{\frac{2}{3}u(n)}_{\text{稳态响应}}$$

暂态响应取决于单位圆内的极点，稳态响应取决于单位圆上的极点。注意，当极点位于单位圆外时，该响应称作无界响应。

- 零输入响应(或初始条件响应)和零状态响应：

在式(4.27)中，$Y^+(z)$由两部分组成。第一部分可以解释为

$$Y_{ZS}(z)=H(z)X(z)$$

第二部分可解释为

$$Y_{ZI}(z)=H(z)X_{IC}(z)$$

其中 $X_{IC}(z)$可以看作等价的初始条件输入，此输入与初始条件产生同样的输出 Y_{ZI}。在此例中，$X_{IC}(n)$是

$$X_{IC}(n)=\{\underset{\uparrow}{1},-2\}$$

对(4.27)式的两部分同时作 z 反变换，全响应可写为

$$y(n)=\underbrace{\left[\frac{1}{3}\left(\frac{1}{4}\right)^n-2\left(\frac{1}{2}\right)^n+\frac{8}{3}\right]u(n)}_{\text{零状态响应}}+\underbrace{\left[3\left(\frac{1}{2}\right)^n-2\right]u(n)}_{\text{零输入响应}}$$

从本例可以清楚地看出，一般来讲全响应的每一部分都是不同的函数，并且分别侧重于系统分析的不同方面。

MATLAB 实现

在第二章中我们用 filter 函数来解差分方程,需要给出方程的系数和输入。当初始条件给出时,此函数也可用于求全响应。此时,filter 用如下格式调用

```
y = filter(b,a,x,xic)
```

其中,xic 是等效的初始条件输入序列。要计算例 4.14 的全响应,可以这样做:

```
>> n = [0:7]; x = (1/4).^n; xic = [1, -2];
>> format long
>> y1 = filter(b,a,x,xic)
y1 =
  Columns 1 through 4
    2.00000000000000   1.25000000000000   0.93750000000000   0.79687500000000
  Columns 5 through 8
    0.73046875000000   0.69824218750000   0.68237304687500   0.67449951171875
>> y2 = (1/3) * (1/4).^n + (1/2).^n + (2/3) * ones(1,8) %MATLAB 验算
y2 =
  Columns 1 through 4
    2.00000000000000   1.25000000000000   0.93750000000000   0.79687500000000
  Columns 5 through 8
    0.73046875000000   0.69824218750000   0.68237304687500   0.67449951171875
```

它与(4.29)式得出的响应一致。在例 4.14 中,我们解析地计算 $x_{IC}(n)$。然而在实际中,特别是对于高阶差分方程,解析地计算 $x_{IC}(n)$是很复杂的。MATLAB 提供了 filtic 函数,它在信号处理工具箱(2.0b 或更高版本)中。其调用格式为

```
xic = filtic(b,a,Y,X)
```

其中,b 和 a 是滤波器的系数数组,Y 和 X 是初始条件数组,分别由 $y(n)$和 $x(n)$的初始条件确定,其形式如下

$$Y = [y(-1), y(-2), \ldots, y(-N)]$$
$$X = [x(-1), x(-2), \ldots, x(-M)]$$

如果 $n \leq -1$ 时 $x(n) = 0$,则 filtic 函数中不用指定 X。对于例 4.14,我们可以这样用

```
>> Y = [4,10];
```

```
>>xic=filtic(b,a,Y)
xic =
     1   -2
```

来确定 $x_{IC}(n)$。

例4.15 求解差分方程

$$y(n)=\frac{1}{3}[x(n)+x(n-1)+x(n-2)]+0.95y(n-1)-0.9025y(n-2),\ n\geq 0$$

其中 $x(n)=\cos(\pi n/3)u(n)$,且

$$y(-1)=-2;y(-2)=-3;x(-1)=1;x(-2)=1$$

先解析地求解,然后用 MATLAB 计算。

解:对差分方程作单边 z 变换

$$\begin{aligned}Y^{+}(z)&=\frac{1}{3}[X^{+}(z)+x(-1)+z^{-1}X^{+}(z)+x(-2)+z^{-1}x(-1)+z^{-2}X^{+}(z)]\\&=0.95[y(-1)+z^{-1}Y^{+}(z)]-0.9025[y(-2)+z^{-1}y(-1)+z^{-2}Y^{+}(z)]\end{aligned}$$

代入初始条件,得到

$$Y^{+}(z)=\frac{\frac{1}{3}+\frac{1}{3}z^{-1}+\frac{1}{3}z^{-2}}{1-0.95z^{-1}+0.9025z^{-2}}X^{+}(z)+\frac{1.4742+2.1383z^{-1}}{1-0.95z^{-1}+0.9025z^{-2}}$$

显然,$x_{IC}(n)=[1.4742,\ 2.1383]$。现在代入 $X^{+}(z)=\frac{1-0.5z^{-1}}{1-z^{-1}+z^{-2}}$ 并化简,得到 $Y^{+}(z)$,它是个有理函数。化简和进一步的部分分式展开可以用 MATLAB 来做,程序如下:

```
>>b=[1,1,1]/3; a=[1,-0.95,0.9025];
>>Y=[-2,-3]; X=[1,1];
>>xic=filtic(b,a,Y,X)
xic =
     1.4742      2.1383
>>bxplus=[1,-0.5]; axplus=[1,-1,1]; %X(z)的变换系数
>>ayplus=conv(a,axplus) %Yplus(z)的分母
ayplus =
     1.0000   -1.9500   2.8525   -1.8525   0.9025
>>byplus=conv(b,bxplus)+conv(xic,axplus) %Yplus(z)的分子
```

```
byplus =
    1.8075    0.8308    -0.4975    1.9717
>>[R,p,C] = residuez(byplus,ayplus)
R =
    0.0584 + 3.9468i  0.0584 - 3.9468i  0.8453 + 2.0311i  0.8453 - 2.0311i
p =
    0.5000 - 0.8660i  0.5000 + 0.8660i  0.4750 + 0.8227i  0.4750 - 0.8227i
C =
    []
>>Mp = abs(p), Ap = angle(p)/pi %极坐标形式
Mp =
    1.0000   1.0000   0.9500   0.9500
Ap =
    0.3333   -0.3333   0.3333   -0.3333
```

因此

$$
\begin{aligned}
Y^+(z) &= \frac{1.8076+0.8308z^{-1}-0.4975z^{-2}+1.9717z^{-3}}{1-1.95z^{-1}+2.8525z^{-2}-1.8525z^{-3}+0.9025z^{-4}} \\
&= \frac{0.0584+j3.9468}{1-e^{-j\pi/3}z^{-1}} + \frac{0.0584-j3.9468}{1-e^{j\pi/3}z^{-1}} \\
&\quad + \frac{0.8453+j2.0311}{1-0.95e^{j\pi/3}z^{-1}} + \frac{0.8453-j2.0311}{1-0.95e^{-j\pi/3}z^{-1}}
\end{aligned}
$$

根据表 4.1

$$
\begin{aligned}
y(n) &= (0.0584+j3.9468)e^{-j\pi n/3} + (0.0584-j3.9468)e^{j\pi n/3} \\
&\quad + (0.8453+j2.031)(0.95)^n e^{j\pi n/3} + (0.8453-j2.031)(0.95)^n e^{-j\pi n/3} \\
&= 0.1169\cos(\pi n/3) + 7.8937\sin(\pi n/3) \\
&\quad + (0.95)^n[1.6906\cos(\pi n/3) - 4.0623\sin(\pi n/3)], \quad n \geq 0
\end{aligned}
$$

$y(n)$的前两项对应稳态响应,也就是特解,后两项是暂态响应,也就是通解。

要用 MATLAB 解此题,需要用 filtic 函数,前面我们已用它求 $x_{IC}(n)$序列。得到的解是数值型的。下面计算 $y(n)$的前 8 个点。

```
>>n = [0:7]; x = cos(pi * n/3);
>>y = filter(b,a,x,xic)
y =
  Columns 1 through 4
    1.80750000000000   4.35545833333333   2.83975000000000   -1.56637197916667
```

```
  Columns 5 through 8
  -4.71759442187500   -3.40139732291667   1.35963484230469   5.02808085078841
% MATLAB 验算
>> A = real(2 * R(1)); B = imag(2 * R(2)); C = real(2 * R(3)); D = imag(2 * R(4));
>> y = A * cos(pi * n/3) + B * sin(pi * n/3) + ((0.95).^n) .* (C * cos(pi * n/3) + D * sin
(pi * n/3))
y =
  Columns 1 through 4
    1.80750000000048   4.35545833333359   2.83974999999978   -1.56637197916714
  Columns 5 through 8
  -4.71759442187528   -3.40139732291648   1.35963484230515   5.02808085078871
```

习　题

4.1　用定义式(4.1)求下列序列的 z 变换。写出每个序列的收敛域,并用 MATLAB 验算得出的 z 变换表达式。

a. $x(n)=\{3,2,1,-2,-3\}$

b. $x(n)=(0.8)^n u(n-2)$

c. $x(n)=\left(\frac{4}{3}\right)^n u(1-n)$

d. $x(n)=2^{-|n|}+\left(\frac{1}{3}\right)^{|n|}$

e. $x(n)=(n+1)(3)^n u(n)$

4.2　利用 z 变换表和 z 变换的性质,求下列序列的 z 变换。$X(z)$写成 z^{-1}的有理函数。用 MATLAB 验算所得结果。分别写出其收敛域,并画出零-极点图。

a. $x(n)=2\delta(n-2)+3u(n-3)$

b. $x(n)=\left(\frac{1}{3}\right)^n u(n-2)+(0.9)^{n-3}u(n)$

c. $x(n)=n\sin\left(\frac{\pi n}{3}\right)u(n)+(0.9)^n u(n-2)$

d. $x(n)=\left(\frac{1}{2}\right)^n\cos\left(\frac{\pi n}{4}-45^\circ\right)u(n-1)$

e. $x(n)=(n-3)\left(\frac{1}{4}\right)^{n-2}\cos\left\{\frac{\pi}{2}(n-1)\right\}u(n)$

4.3　已知 $x(n)$的 z 变换是$X(z)=(1+2z^{-1})$,$|z|\neq0$。求下列序列的 z 变换,并写出其收敛域。

a. $x_1(n)=x(3-n)+x(n-3)$

b. $x_2(n)=(1+n+n^2)x(n)$

c. $x_3(n)=\left(\frac{1}{2}\right)^n x(n-2)$

d. $x_4(n)=x(n+2)*x(n-2)$

e. $x_5(n)=\cos(\pi n/2)x^*(n)$

4.4 重做4.3题,如果

$$X(z)=\frac{1+z^{-1}}{1+\frac{5}{6}z^{-1}+\frac{1}{6}z^{-2}};\ |z|>\frac{1}{2}$$

4.5 已知 $X(z)$的反变换为 $x(n)=(0.5)^n u(n)$。利用 z 变换的性质,分别计算出以下序列 $x(n)$。

a. $X_1(z)=\frac{z-1}{z}X(z)$

b. $X_2(z)=zX(z^{-1})$

c. $X_3(z)=2X(3z)+3X(z/3)$

d. $X_4(z)=X(z)X(z^{-1})$

e. $X_5(z)=z^2\frac{dX(z)}{dz}$

4.6 如果序列 $x_1(n)$、$x_2(n)$和 $x_3(n)$满足 $x_3(n)=x_1(n)*x_2(n)$,那么

$$\sum_{n=-\infty}^{\infty}x_3(n)=\left(\sum_{n=-\infty}^{\infty}x_1(n)\right)\left(\sum_{n=-\infty}^{\infty}x_2(n)\right)$$

a. 在左边代入卷积的定义式,证明上面的结果。

b. 利用卷积的性质证明上面的结果。

c. 选取两个随机序列 $x_1(n)$和 $x_2(n)$,用 MATLAB 验证上面的结果。

4.7 用 MATLAB 计算下列多项式运算的结果

a. $X_1(z)=(1-2z^{-1}+3z^{-2}-4z^{-3})(4+3z^{-1}-2z^{-2}+z^{-3})$

b. $X_2(z)=(z^2-2z+3+2z^{-1}+z^{-2})(z^3-z^{-3})$

c. $X_3(z)=(1+z^{-1}+z^{-2})^3$

d. $X_4(z)=X_1(z)X_2(z)+X_3(z)$

e. $X_5(z)=(z^{-1}-3z^{-3}+2z^{-5}+5z^{-7}-z^{-9})(z+3z^2+2z^3+4z^4)$

4.8 函数 deconv 用于拆分两个因果序列。写一段 MATLAB 函数 deconv _ m 来拆分两个非因果序列(类似于函数 conv)。该函数格式应该如下:

```
function  [p,np,r,nr] = deconv_m(b,nb,a,na)
% 修改后的用于非因果序列解卷积的子程序
%function  [p,np,r,nr] = deconv_m(b,nb,a,na)
%
% p = 多项式部分,基底为 np1 <= n <= np2
% np = [np1,np2]
% r = 余项,基底为 nr1 <= n <= nr2
%nr = [nr1,nr2]
```

```
% b=分子多项式,基底为 nb1 < = n < = nb2
%nb=[nb1,nb2]
% a=分母多项式,基底为 na1 < = n < = na2
%na=[na1,na2]
%
```

用下面的运算检验所编函数

$$\frac{z^2+z+1+z^{-1}+z^{-2}+z^{-3}}{z+2+z^{-1}}=(z-1+2z^{-1}-2z^{-2})+\frac{3z^{-2}+3z^{-3}}{z+2+z^{-1}}$$

4.9 用部分分式展开法求下列 z 反变换。

a. $X_1(z)=(1-z^{-1}-4z^{-2}+4z^{-3})/(1-\frac{11}{4}z^{-1}+\frac{13}{8}z^{-2}-\frac{1}{4}z^{-3})$. 该序列是右序列。

b. $X_2(z)=(1-z^{-1}-4z^{-2}+4z^{-3})/(1-\frac{11}{4}z^{-1}+\frac{13}{8}z^{-2}-\frac{1}{4}z^{-3})$. 该序列是绝对可加的。

c. $X_3(z)=(z^3-3z^2+4z+1)/(z^3-4z^2+z-0.16)$. 该序列是左序列。

d. $X_4(z)=z/(z^3+2z^2+1.25z+0.25)$, $|z|>1$.

e. $X_5(z)=z/(z^2-0.25)^2$, $|z|<0.5$.

4.10 假设 $X(z)$如下

$$X(z)=\frac{2+3z^{-1}}{1-z^{-1}+0.81z^{-2}},\ |z|>0.9$$

a.求出 $x(n)$,使其不包含复数项。

b.用 MATLAB 计算 $x(n)$的前 20 个点,并与(a)的结果比较。

4.11 对于下列由脉冲响应描述的线性时不变系统,求(ⅰ)系统函数表达式;(ⅱ)差分方程表达式;(ⅲ)零-极点图;(ⅳ)输入 $x(n)=(0.25)^n u(n)$时的输出 $y(n)$。

a. $h(n)=2\left(\frac{1}{2}\right)^n u(n)$

b. $h(n)=n\left(\frac{1}{3}\right)^n u(n)+\left(-\frac{1}{4}\right)^n u(n)$

c. $h(n)=3(0.9)^n\cos(\pi n/4+\pi/3)u(n+1)$

d. $h(n)=n[u(n)-u(n-10)]$

e. $h(n)=[2-\sin(\pi n)]u(n)$

4.12 对于由下列系统函数描述的线性时不变系统,求(ⅰ)脉冲响应表达式;(ⅱ)差分方程表达式;(ⅲ)零-极点图;(ⅳ)输入 $x(n)=3\cos(\pi n/3)u(n)$时的输出 $y(n)$。

a. $H(z)=(z+1)/(z-0.5)$, 因果系统。

b. $H(z)=(1+z^{-1}+z^{-2})/(1+0.5z^{-1}-0.25z^{-2})$, 稳定系统。

c. $H(z)=(z^2-1)/(z-3)^2$,非因果系统。

d. $H(z)=\frac{z}{z-0.25}+\frac{1-0.5z^{-1}}{1+2z^{-1}}$, 稳定系统。

e. $H(z)=(1+z^{-1}+z^{-2})^2$

4.13 对于下列由差分方程描述的线性、因果、时不变系统，求(ⅰ)脉冲响应表达式；(ⅱ)系统函数表达式；(ⅲ)零-极点图；(ⅳ)输入 $x(n)=2(0.9)^n u(n)$时的输出 $y(n)$。

a. $y(n)=[x(n)+2x(n-1)+x(n-3)]/4$

b. $y(n)=x(n)+0.5x(n-1)-0.5y(n-1)+0.25y(n-2)$

c. $y(n)=2x(n)+0.9y(n-1)$

d. $y(n)=-0.45x(n)-0.4x(n-1)+x(n-2)+0.4y(n-1)+0.45y(n-2)$

e. $y(n)=\sum_{m=0}^{4}(0.8)^m x(n-m)-\sum_{l=1}^{4}(0.9)^l y(n-l)$

4.14 习题4.13的输出序列 $y(n)$是全响应。对于题中的每个系统，把 $y(n)$分成(ⅰ)通解；(ⅱ)特解；(ⅲ)暂态响应；(ⅳ)稳态响应。

4.15 已知一个稳定系统的零-极点分布如下：

$$z_1=j,\ z_2=-j,\ p_1=-\frac{1}{2}+j\frac{1}{2},\ p_2=-\frac{1}{2}-j\frac{1}{2}$$

还知道其频率响应函数 $H(e^{j\omega})$在 $\omega=0$ 处的值为0.8；即 $H(e^{j0})=0.8$

a. 求系统函数 $H(z)$并写出收敛域。

b. 求差分方程表达式。

c. 求输入 $x(n)=\frac{1}{\sqrt{2}}\sin\left(\frac{\pi n}{2}\right)u(n)$时的稳态响应 $y_{ss}(n)$。

d. 求输入 $x(n)=\frac{1}{\sqrt{2}}\sin\left(\frac{\pi n}{2}\right)u(n)$时的暂态响应 $y_{tr}(n)$。

4.16 一个数字滤波器的差分方程如下

$$y(n)=x(n)+x(n-1)+0.9y(n-1)-0.81y(n-2)$$

a、用 freqz 函数画出该滤波器的幅频和相频响应曲线，注意在 $\omega=\pi/3$ 和 $\omega=\pi$ 时的幅度和相位值。

b.产生信号 $x(n)=\sin(\pi n/3)+5\cos(\pi n)$的200个点并使其通过滤波器。把输出的稳态部分与 $x(n)$比较，讨论滤波器如何影响两个正弦波的幅度和相位?

4.17 用单边 z 变换解下面的差分方程，求 $y(n)$。

$$y(n)=0.5y(n-1)+0.25y(n-2)+x(n),\ n\geq 0;\ y(-1)=1,\ y(-2)=2$$
$$x(n)=(0.8)^n u(n)$$

用 MATLAB 解出 $y(n)$的前20个点并与你的答案比较。

4.18 解差分方程，求 $y(n)$，$n\geq=0$

$$y(n)-0.4y(n-1)-0.45y(n-2)=0.45x(n)+0.4x(n-1)-x(n-2)$$

设输入 $x(n)=2+(0.5)^n u(n)$，初始条件为

$$y(-1)=0,\ y(-2)=3;\ x(-1)=x(-2)=2$$

把解 $y(n)$分解成(ⅰ)暂态响应;(ⅱ)稳态响应;(ⅲ)零输入响应;(ⅳ)零状态响应。

4.19 已知一个因果、线性、时不变系统由下面的差分方程描述:

$$y(n)=y(n-1)+y(n-2)+x(n-1)$$

a.求该系统的系统函数 $H(z)$。

b.画出 $H(z)$的零、极点并指出收敛域。

c.求该系统的单位脉冲响应 $h(n)$。

d.该系统是否稳定? 若稳定,试证明之;若不稳定,找出一个满足该差分方程的稳定的单位脉冲响应。

4.20 求如下系统的零状态响应

$$y(n)=0.25y(n-1)+x(n)+3x(n-1),\ n\geq 0;\ y(-1)=2$$

对于输入

$$x(n)=e^{j\pi n/4}u(n)$$

该系统的稳态响应是什么?

第五章　离散付利叶变换

在第三章和第四章中我们学习了离散信号在两种域中变换的表示方法。离散时间付利叶变换提供了绝对可加序列在频域(ω)中的表示方法。z 变换则提供任意序列的频域表示。这两种变换有两个共同特征。第一,变换适用于无限长序列,第二,也是最重要的一点,它们是连续变量(ω 或 z)的函数。从数值可计算的角度来看(或从 MATLAB 的角度看),这些特征是不利的,因为这要计算无限长序列的无限项和。为了使用 MATLAB 我们必须截断序列,而得到有限个点的表达式。这在前两章中的许多例子中已使用过。显然,这种估算是对精确计算的一种近似。换句话说,离散时间付利叶变换和 z 变换均是不能进行数值计算的。

因此我们将注意力转向可数值计算的变换。它由对离散时间付利叶变换在频域中(或 z 变换的单位圆上)采样所得。首先通过分析周期序列来研究这种变换。由付利叶分析可知,一个周期函数可由其各谐波分量的线性组合得到,其采样形式为复指数形式。这就是我们所学的离散付利叶级数(DFS)表示法。由于是在频域中进行的采样,因此需要研究采样对时域的影响,以及在 z 域中重构的结果。随后将 DFS 推广至有限持续时间序列,产生一个新的变换,称为离散付利叶变换(或 DFT)。DFT 避免了前面所提到的那两个问题,并且它是计算机可实现的数值计算的变换。我们将详细讨论它的特性以及在系统分析中的应用。长序列的 DFT 数值计算将耗时甚多,因此我们将探讨几种计算 DFT 的有效算法,统称为快速付利叶变换(或 FFT)算法。我们将详细讨论其中的两种算法。

离散付利叶级数

第二章中我们曾经用 $\tilde{x}(n)$ 定义周期序列,它满足以下条件:

$$\tilde{x}(n)=\tilde{x}(n+kN),\ \forall n,k \tag{5.1}$$

其中 N 是序列的基本周期。由付利叶分析我们知道周期函数可由复指数的线性组合叠加得到。其频率为基本频率(在这里为 $2\pi/N$)的倍数(或谐波)。从离散时间付利叶变换的频域周期性,我们知道谐波次数是有限的,其频率为 $\{\frac{2\pi}{N}k, k=0,1,\cdots,N-1\}$。因此周期序列 $\tilde{x}(n)$ 可表示成:

$$\tilde{x}(n)=\frac{1}{N}\sum_{k=0}^{N-1}\tilde{X}(k)e^{j\frac{2\pi}{N}kn},\ n=0,\pm1,\ldots, \tag{5.2}$$

其中 $\{\tilde{X}(k), k=0,\pm1,\ldots,\}$ 叫做离散付利叶级数系数,由下式给出:

$$\tilde{X}(k)=\sum_{n=0}^{N-1}\tilde{x}(n)e^{-j\frac{2\pi}{N}nk},\ k=0,\pm1,\ldots, \tag{5.3}$$

注意 $\tilde{X}(k)$本身是一个基本周期为 N 的(复值)周期序列,即:

$$\tilde{X}(k+N)=\tilde{X}(k) \tag{5.4}$$

一对等式(5.2)和(5.3)合在一起称为周期序列的离散付利叶级数表示。采用 $W_N \triangleq e^{-j\frac{2\pi}{N}}$ 表示复指数,我们将(5.2)和(5.3)表示为:

$$\begin{aligned} \tilde{X}(k) &\triangleq \text{DFS}[\tilde{x}(n)] = \sum_{n=0}^{N-1} \tilde{x}(n) W_N^{nk} && \text{分析或 DFS 等式} \\ \tilde{x}(n) &\triangleq \text{IDFS}[\tilde{X}(k)] = \frac{1}{N} \sum_{k=0}^{N-1} \tilde{X}(k) W_N^{-nk} && \text{综合或逆 DFS 等式} \end{aligned} \tag{5.5}$$

例 5.1 求出下面周期序列的 DFS 表示式:

$$\tilde{x}(n)=\{\ldots,0,\underset{\uparrow}{1},2,3,0,1,2,3,0,1,2,3,\ldots\}$$

解:上述序列的基本周期为 $N=4$,因而 $W_4=e^{-j\frac{2\pi}{4}}=-j$,现在

$$\tilde{X}(k)=\sum_{n=0}^{3} \tilde{x}(n) W_4^{nk}, \ k=0, \pm 1, \pm 2, \ldots$$

因而

$$\tilde{X}(0)=\sum_{0}^{3} \tilde{x}(n) W_4^{0\cdot n}=\sum_{0}^{3} \tilde{x}(n)=\tilde{x}(0)+\tilde{x}(1)+\tilde{x}(2)+\tilde{x}(3)=6$$

类似地

$$\begin{aligned} \tilde{X}(1) &= \sum_{0}^{3} \tilde{x}(n) W_4^{n} = \sum_{0}^{3} \tilde{x}(n)(-j)^{n} = (-2+2j) \\ \tilde{X}(2) &= \sum_{0}^{3} \tilde{x}(n) W_4^{2n} = \sum_{0}^{3} \tilde{x}(n)(-j)^{2n} = 2 \\ \tilde{X}(3) &= \sum_{0}^{3} \tilde{x}(n) W_4^{3n} = \sum_{0}^{3} \tilde{x}(n)(-j)^{3n} = (-2-2j) \end{aligned}$$

MATLAB 实现

仔细观察式(5.5)发现 DFS 是一种可数值计算表示式,它可由多种方式来实现。为了计算每个样本 $\tilde{X}(k)$,我们可用 for...end 环来实现求和。为计算所有的 DFS 系数,需要另外一

个 for...end 环，这将导致运行嵌套的两个 for...end 环。显然，这在 MATLAB 中效率是很低的。一种有效的 MATLAB 实现是对(5.5)的每一个关系式采用矩阵—矢量乘法。我们早先在实现离散时间付利叶变换的数值逼近时，曾采用过此方法。设 $\tilde{x}$ 和 $\tilde{X}$ 代表序列 $x(n)$ 和 $X(k)$ 主周期的列向量。则(5.5)由下式给出：

$$
\begin{aligned}
\tilde{X} &= W_N \tilde{x} \\
\tilde{x} &= \frac{1}{N} W_N^* \tilde{X}
\end{aligned}
\tag{5.6}
$$

其中矩阵 W_N 由下式给出：

$$
W_N \triangleq [W_N^{kn}]_{0 \le k,n \le N-1} = k \downarrow \overset{n \longrightarrow}{\begin{bmatrix} 1 & 1 & \cdots & 1 \\ 1 & W_N^1 & \cdots & W_N^{N-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & W_N^{N-1} & \cdots & W_N^{(N-1)^2} \end{bmatrix}}
\tag{5.7}
$$

矩阵 W_N 为方阵，叫做 DFS 矩阵。可用下面的 MATLAB 函数 dfs 实现上述过程：

```
function [Xk] = dfs(xn,N)
% 计算离散付利叶级数(DFS)系数
% ------------------------------
% [Xk] = dfs(xn,N)
% Xk = 在 0 <= k <= N-1 间的 DFS 系数数组
% xn = 周期信号在 0 <= n <= N-1 之间的一个单周期信号
% N = xn 的基本周期
%
n = [0:1:N-1];               % n 的行向量
k = [0:1:N-1];               % k 的行向量
WN = exp(-j*2*pi/N);         % Wn 因子
nk = n'*k;                   % 产生一个含 nk 值的 N 乘 N 维矩阵
WNnk = WN .^ nk;             % DFS 矩阵
Xk = xn * WNnk;              % DFS 系数的行向量
```

例 5.1 中的 DFS 可由 MATLAB 来计算：

```
>> xn = [0,1,2,3]; N = 4;
>> xk = dfs(xn,N)
xk =
    6.0000   -2.0000 + 2.0000i   -2.0000 - 0.0000i   -2.0000 - 2.0000i
```

下面的 idfs 函数实现合成式：

```
function [xn] = idfs(Xk,N)
% 计算逆离散付利叶级数(IDFS)
% ————————————————————
% [xn] = idfs(Xk,N)
% xn = 周期信号在 0 < = n < = N-1 之间的一个单周期信号
% Xk = 在 0 < = k < = N-1 间的 DFS 系数数组
% N = Xk 的基本周期
%
n = [0:1:N-1];              % n 的行向量
k = [0:1:N-1];              % k 的行向量
WN = exp(-j*2*pi/N);        % Wn 因子
nk = n'*k;                  % 产生一个含 nk 值的 N 乘 N 维矩阵
WNnk = WN .^ (-nk);         % IDFS 矩阵
xn = (Xk * WNnk)/N;         % IDFS 值的行向量
```

注意：以上函数是用 MATLAB 实现(5.5)的有效方法，但它们的计算效率不高，尤其是对于大的 N 值。稍后我们将在这章中讨论这个问题。

例 5.2 下面给出一周期"方波"序列：

$$\tilde{x}(n)=\begin{cases}1, & mN\le n\le mN+L-1\\ 0, & mN+L\le n\le (m+1)N-1\end{cases};\ m=0,\pm 1,\pm 2,\ldots$$

其中 N 是基本周期，L/N 是占空比。

a. 确定一种用 L 与 N 描述的 $|\tilde{X}(k)|$ 的表达式。

b. 分别画出当 $L=5, N=20$；$L=5, N=40$；$L=5, N=60$；$L=7, N=60$ 时 $|\tilde{X}(k)|$ 的幅值。

c. 对所得结果进行讨论。

解：图 5.1 给出当 $L=5, N=20$ 时，此序列的图

a. 用分析式(5.3)可得：

$$\tilde{X}(k)=\sum_{n=0}^{N-1}\tilde{x}(n)e^{-j\frac{2\pi}{N}nk}=\sum_{n=0}^{L-1}e^{-j\frac{2\pi}{N}nk}=\sum_{n=0}^{L-1}\left(e^{-j\frac{2\pi}{N}k}\right)^{n}$$

$$=\begin{cases}L, & k=0,\pm N,\pm 2N,\ldots\\ \dfrac{1-e^{-j2\pi Lk/N}}{1-e^{-j2\pi k/N}}, & \text{其它}\end{cases}$$

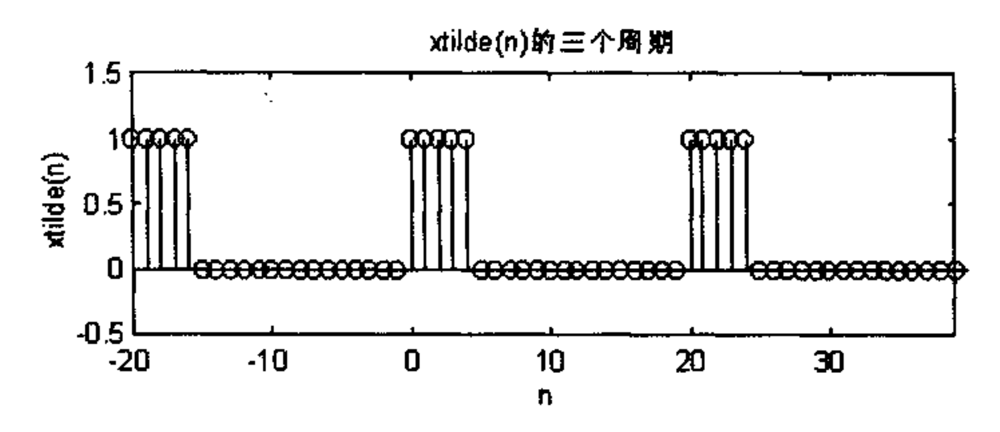


图 5.1 周期方波序列

最后一步是由第二章的几何级数求和公式(2.5)得到的,它可简化为:

$$\frac{1-e^{-j2\pi Lk/N}}{1-e^{-j2\pi k/N}}=\frac{e^{-j\pi Lk/N}}{e^{-j\pi k/N}}\,\frac{e^{-j\pi Lk/N}-e^{-j\pi Lk/N}}{e^{-j\pi k/N}-e^{-j\pi k/N}}$$

$$=e^{-j\pi(L-1)k/N}\frac{\sin(\pi kL/N)}{\sin(\pi k/N)}$$

$\tilde{X}(k)$的幅值还可表示为:

$$|\tilde{X}(k)|=\begin{cases}L, & k=0,\pm N,\pm 2N,\dots\\ \left|\dfrac{\sin(\pi kL/N)}{\sin(\pi k/N)}\right|, & \text{其它}\end{cases}$$

b. 下面给出 $L=5, N=20$ 的 MATLAB 程序:

```
>> L=5; N=20; k=[-N/2:N/2];% 方波参数
>> xn=[ones(1,L), zeros(1,N-L)]; % 方波 x(n)
>> Xk=dfs(xn,N); % 离散付利叶级数 DFS
>> magXk=abs([Xk(N/2+1:N) Xk(1:N/2+1)]); % DFS 幅度
>> subplot(2,2,1); stem(k,magXk); axis([-N/2,N/2,-0.5,5.5])
>> xlabel('k') ylabel('Xtilde(k)');
>> title('方波的 DFS: L=5, N=20')
```

上述程序所产生的图以及其他情形的图如图 5.2 所示。注意到 $\tilde{X}(k)$是周期信号,图中只画出了从 $-N/2$ 到 $N/2$ 的部分。

c. 从图 5.2 中可得到一些有趣的结论。方波的 DFS 系数的包络看起来像“sinc”函数。$k=0$ 时的幅度为 L,同时函数的零点位于 N/L(占空比的倒数)的整数倍处。稍后我们将在这章中研究这些函数。

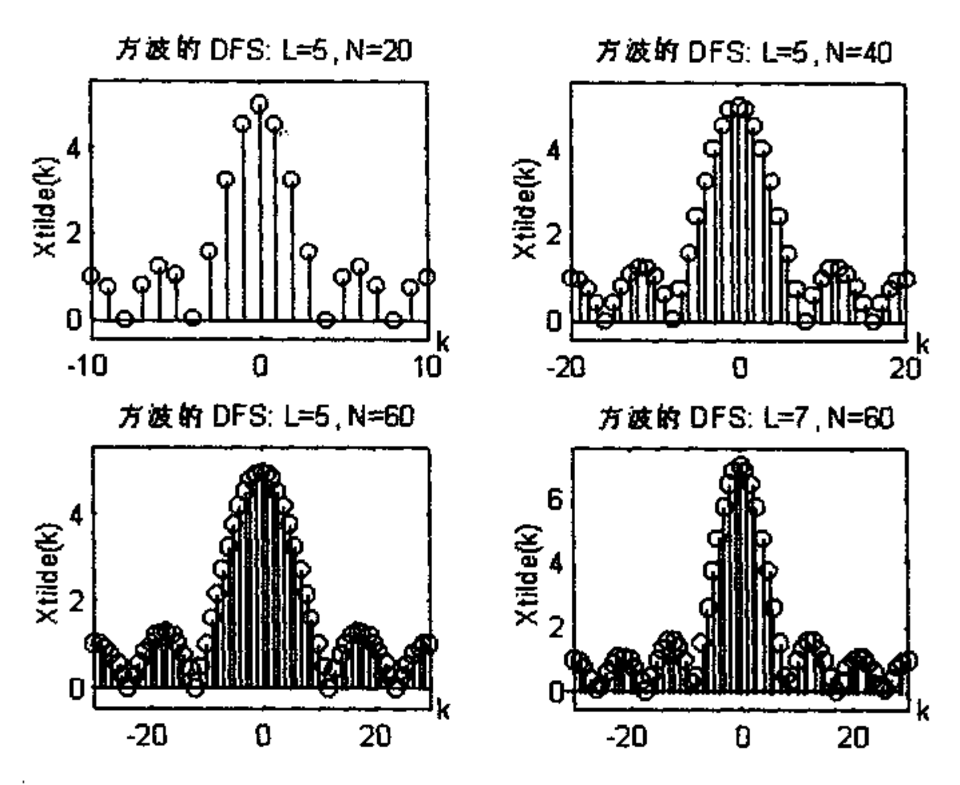


图 5.2　不同 L 和 N 的周期方波的离散付利叶级数幅值

与 z 变换的关系

设 $x(n)$ 为长度为 N 的有限时间序列：

$$x(n)=\begin{cases}\text{非零},0\leq n\leq N-1\\ 0,\quad \text{其它}\end{cases} \tag{5.8}$$

则可得到它的 z 变换：

$$X(z)=\sum_{n=0}^{N-1}x(n)z^{-n} \tag{5.9}$$

现在我们以周期 N 重复 $x(n)$，构造一个周期序列 $\tilde{x}(n)$，即：

$$x(n)=\begin{cases}\tilde{x}(n),0\leq n\leq N-1\\ 0,\quad\quad \text{其它}\end{cases} \tag{5.10}$$

$\tilde{x}(n)$的 DFS 由下式给出：

$$\tilde{X}(k)=\sum_{n=0}^{N-1}\tilde{x}(n)e^{-j\frac{2\pi}{N}nk}=\sum_{n=0}^{N-1}x(n)[e^{j\frac{2\pi}{N}k}]^{-n} \tag{5.11}$$

将它与(5.9)比较,我们得到:

$$\tilde{X}(k)=X(z)|_{z=e^{j\frac{2\pi}{N}k}} \tag{5.12}$$

这意味着 DFS$\tilde{X}(k)$表示 z 变换 $X(z)$在单位圆上的 N 个等间隔样本。

与 DTFT 的关系

既然(5.8)中的 $x(n)$是长度为 N 的有限持续时间序列,它也是绝对可加的,因而其 DTFT 存在,并由下式给出:

$$X(e^{j\omega})=\sum_{n=0}^{N-1}x(n)e^{-j\omega n}=\sum_{n=0}^{N-1}\tilde{x}(n)e^{-j\omega n} \tag{5.13}$$

将(5.13)与(5.11)比较,得到:

$$\tilde{X}(k)=X(e^{j\omega})|_{\omega=\frac{2\pi}{N}k} \tag{5.14}$$

使

$$\omega_1\triangleq\frac{2\pi}{N}\quad 与\quad \omega_k\triangleq\frac{2\pi}{N}k=k\omega_1$$

则 DFS $X(k)=X(e^{j\omega k})=X(e^{jk\omega_1})$,这就意味着 DFS 可由对 DTFT 以 $\omega_1=\frac{2\pi}{N}$为间隔进行等间隔采样得到。观察(5.12)与(5.14),可知 DFS 表示式提供了一种在频域中的采样技巧,其原理上,与时域中的采样类似。间隔 $\omega_1=\frac{2\pi}{N}$称为频域中的采样间隔。由于它也告诉我们频率采样(或测量)的接近程度,因此它也称为频率分辨率。

例 5.3　$x(n)=\{\underset{\uparrow}{0},1,2,3\}$

a. 计算它的离散时间付利叶变换 $X(e^{j\omega})$

b. 对 $X(e^{j\omega})$以 $k\omega_1=\frac{2\pi}{4}k,k=0,1,2,3$ 进行采样,证明它与例 5.1 中的 $\tilde{X}(k)$相等。

解:序列 $x(n)$不是周期序列,但它的持续时间有限。

a. 离散时间付利叶变换由下式给出:

$$X(e^{j\omega})=\sum_{n=-\infty}^{\infty}x(n)e^{-j\omega n}=e^{-j\omega}+2e^{-j2\omega}+3e^{-j3\omega}$$

b. 以 $k\omega_1=\frac{2\pi}{N}k,k=0,1,2,3$ 进行采样,我们得到:

$$X(e^{j0}) = 1+2+3 = 6 = \tilde{X}(0)$$
$$X(e^{j2\pi/4}) = e^{-j2\pi/4} + 2e^{-j4\pi/4} + 3e^{-j6\pi/4} = -2+2j = \tilde{X}(1)$$
$$X(e^{j4\pi/4}) = e^{-j4\pi/4} + 2e^{-j8\pi/4} + 3e^{-j12\pi/4} = -2 = \tilde{X}(2)$$
$$X(e^{j6\pi/4}) = e^{-j6\pi/4} + 2e^{-j12\pi/4} + 3e^{-j18\pi/4} = -2-2j = \tilde{X}(3)$$

与预想的一样。

z 域中的采样与重构

假设 $x(n)$是具有有限持续时间的任意可加序列，它的 z 变换为：

$$X(z) = \sum_{m=-\infty}^{\infty} x(m) z^{-m}$$

同时我们假设 $X(z)$的 ROC 包括单位圆，我们以 $\omega_1 = 2\pi/N$ 对 $X(z)$在单位圆上进行等角度采样，并称它为 DFS 序列。

$$\tilde{X}(k) \triangleq X(z)|_{z=e^{j\frac{2\pi}{N}k}}, \qquad k = 0, \pm 1, \pm 2, \ldots \tag{5.15}$$
$$= \sum_{m=-\infty}^{\infty} x(m) e^{-j\frac{2\pi}{N}km} = \sum_{m=-\infty}^{\infty} x(m) W_N^{km}$$

是周期为 N 的周期序列。最后我们计算 $\tilde{X}(k)$的 IDFS，

$$\tilde{x}(n) = \text{IDFS}[\tilde{X}(k)]$$

也是周期为 N 的周期序列。显然，任意序列 $x(n)$和 $\tilde{x}(n)$之间必然有一定的关系。这是一个重要的结果。为了在数值上计算逆 DTFT 或 z 反变换。我们必须处理 $X(z)$沿单位圆采样所得的有限个样本，因此，必须知道这样的采样对时域序列的影响。这关系是容易得到的。

$$\tilde{x}(n) = \frac{1}{N}\sum_{k=0}^{N-1} \tilde{X}(k) W_N^{-kn} \qquad (\text{由}(5.2))$$
$$= \frac{1}{N}\sum_{k=0}^{N-1} \left\{ \sum_{m=-\infty}^{\infty} x(m) W_N^{km} \right\} W_N^{-kn} \qquad (\text{由}(5.15))$$

或

$$\tilde{x}(n) = \sum_{m=-\infty}^{\infty} x(m) \underbrace{\frac{1}{N}\sum_{0}^{N-1} W_N^{-k(n-m)}}_{=\begin{cases}1, & n-m=rN\\ 0, & \text{其它}\end{cases}} = \sum_{m=-\infty}^{\infty} x(m) \sum_{r=-\infty}^{\infty} \delta(n-m-rn)$$

$$= \sum_{r=-\infty}^{\infty} \sum_{m=-\infty}^{\infty} x(m)\delta(n-m-rn)$$

或

$$\tilde{x}(n) = \sum_{r=-\infty}^{\infty} x(n-rN) = \cdots + x(n+N) + x(n) + x(n-N) + \cdots \tag{5.16}$$

这意味着当在单位圆上对 $X(z)$采样时，在时域中我们可得到一周期序列。此序列是原始序列 $x(n)$和它的无限个移位复制品(每一次移位 $\pm N$ 的整数倍)的线性组合，例 5.5 已证明了这一点。由(5.16)可知，如果当 $n<0$ 与 $n \geqslant N$ 时，$x(n)=0$，则时域中不会出现混叠。因而我们能从 $\tilde{x}(n)$中识别和恢复出 $x(n)$，即

$$x(n) = \tilde{x}(n) \qquad 0 \le n \le (N-1)$$

或

$$x(n) = \tilde{x}(n) R_N(n) = \tilde{x}(n) \begin{cases} 1, & 0 \le n \le N \\ 0, & \text{其它} \end{cases}$$

其中 $R_N(n)$是长度为 N 的矩形窗。因此有以下的定理。

定理 1　频率采样定理

若 $x(n)$为时间有限(即有限持续时间)序列，长$[0, N-1]$，则 $X(z)$完全由其在单位圆上的 N 个样本确定。

例 5.4　设 $x_1(n) = \{\underset{\uparrow}{6}, 5, 4, 3, 2, 1\}$，在

$$\omega_k = \frac{2\pi k}{4},\ k = 0, \pm 1, \pm 2, \pm 3, \ldots$$

上对其 DTFT $X_1(e^{j\omega})$进行采样，得到一 DFS 序列 $\tilde{x}_2(k)$，求出序列 $\tilde{x}_2(n)$，使其具有 $\tilde{x}_2(k)$的逆 DFS。

解：毋须计算 DTFT、DFS 或逆 DFS，我们可直接用混叠公式(5.16)计算 $\tilde{x}_2(n)$

$$\tilde{x}_2(n) = \sum_{r=-\infty}^{\infty} x_1(n-4r)$$

$x(4)$混叠入 $x(0)$，$x(5)$混叠入 $x(1)$。因此

$$\tilde{x}_2(n) = \{\ldots, 8, 6, 4, 3, \underset{\uparrow}{8}, 6, 4, 3, 8, 6, 4, 3, \ldots\}$$

例 5.5 设 $x(n)=(0.7)^n u(n)$，当 $N=5,10,20,50$ 时分别对其 z 变换在单位圆上采样，研究不同的 N 对时域的影响。

解：查表 1.1，$x(n)$的 z 变换为：

$$X(n)=\frac{1}{1-0.7z^{-1}}=\frac{z}{z-0.7},\quad |z|>0.7$$

我们可用 MATLAB 来实现采样运算

$$\widetilde{X}(k)=X(z)|_{z=e^{j2\pi k/N}},\ k=0,\pm1,\pm2,\ldots$$

与逆 DFS 计算，以确定相应的时域序列。下面给出 $N=5$ 时的 MATLAB 程序：

```
>> N=5; k=0:1:N-1;% 样本下标
>> wk=2*pi*k/N; zk=exp(j*wk);% z的样本
>> Xk=(zk)./(zk-0.7);% X(z)样本的 DFS
>> xn=real(idfs(Xk,N)); % IDFS
>> xtilde=xn'* ones(1,8); xtilde=(xtilde(:))'; % 周期序列
>> subplot(2,2,1); stem(0:39,xtilde);axis([0,40,-0.1,1.5])
>> xlabel('n');ylabel('xtilde(n)'); title('N=5');
```

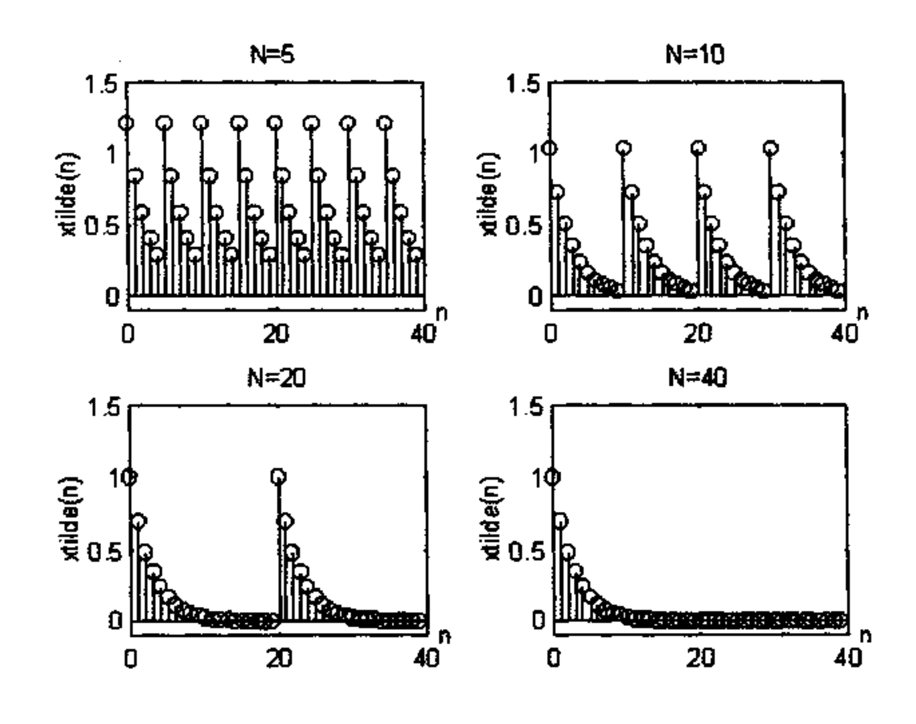


图 5.3 例 5.5 中的图

图 5.3 中的各图清楚地表明在时域中出现的混叠，尤其是当 $N=5$ 与 $N=10$ 时。对于大

的 N 值,其 $x(n)$的尾部足够小,实际上不会导致明显的混叠。这对于变换前,有效截取无限序列,是非常有用的。

重构公式

设 $x(n)$的时宽为$[0,N-1]$,由定理 1 我们可用样本 $\tilde{X}(k)$恢复其 z 变换 $X(z)$。这由下式给出

$$\begin{aligned}X(z) &= Z[x(n)] = Z[\tilde{x}(n)R_N(n)] \\ &= Z[\mathrm{IDFS}\{\underbrace{\tilde{X}(k)}_{X(z)\text{的样本}}\}R_N(n)]\end{aligned}$$

由上述方法可得 z 域重构公式。

$$\begin{aligned}X(z) &= \sum_{0}^{N-1} x(n)z^{-n} = \sum_{0}^{N-1}\tilde{x}(n)z^{-n} \\ &= \sum_{0}^{N-1}\left\{\frac{1}{N}\sum_{0}^{N-1}\tilde{X}(k)W_N^{-kn}\right\}z^{-n} \\ &= \frac{1}{N}\sum_{k=0}^{N-1}\tilde{X}(k)\left\{\sum_{0}^{N-1}W_N^{-kn}z^{-n}\right\} \\ &= \frac{1}{N}\sum_{k=0}^{N-1}\tilde{X}(k)\left\{\sum_{0}^{N-1}(W_N^{-k}z^{-1})^{n}\right\} \\ &= \frac{1}{N}\sum_{k=0}^{N-1}\tilde{X}(k)\left\{\frac{1-W_N^{-kN}z^{-N}}{1-W_N^{-k}z^{-1}}\right\}\end{aligned}$$

因为 $W_N^{-kN}=1$,得到

$$X(z) = \frac{1-z^{-N}}{N}\sum_{k=0}^{N-1}\frac{\tilde{X}(k)}{1-W_N^{-k}z^{-1}} \tag{5.17}$$

DTFT 内插公式

若在单位圆 $z=e^{j\omega}$上用重构公式(5.17),可得出离散付利叶变换,即

$$\begin{aligned}X(e^{j\omega}) &= \frac{1-e^{-j\omega N}}{N}\sum_{k=0}^{N-1}\frac{\tilde{X}(k)}{1-e^{j2\pi k/N}e^{-j\omega}} \\ &= \sum_{k=0}^{N-1}\tilde{X}(k)\frac{1-e^{-j\omega N}}{N\{1-e^{-j2\pi k/N}e^{-j\omega}\}}\end{aligned}$$

考虑到

$$\frac{1-e^{-j\omega N}}{N\{1-e^{j2\pi k/N}e^{-j\omega}\}}=\frac{1-e^{-j\left(\omega-\frac{2\pi k}{N}\right)N}}{N\{1-e^{-j\left(\omega-\frac{2\pi k}{N}\right)}\}}$$

$$=\frac{e^{-j\frac{N}{2}\left(\omega-\frac{2\pi k}{N}\right)}}{e^{-\frac{1}{2}j\left(\omega-\frac{2\pi k}{N}\right)}}\left\{\frac{\sin\left[\left(\omega-\frac{2\pi k}{N}\right)\frac{N}{2}\right]}{N\sin\left[\left(\omega-\frac{2\pi k}{N}\right)\frac{1}{2}\right]}\right\}$$

设

$$\Phi(\omega)\triangleq\frac{\sin\left(\frac{\omega N}{2}\right)}{N\sin\left(\frac{\omega}{2}\right)}e^{-j\omega\left(\frac{N-1}{2}\right)}:\qquad \text{内插多项式} \tag{5.18}$$

则

$$X(e^{j\omega})=\sum_{k=0}^{N-1}\tilde{X}(k)\Phi\left(\omega-\frac{2\pi k}{N}\right) \tag{5.19}$$

这是从 $X(e^{j\omega})$的样本 $\tilde{X}(k)$重构它本身的 DTFT 内插公式。由 $\Phi(0)=1$,得 $X(e^{j2\pi k/N})=\tilde{X}(k)$,这表明在采样点上的内插是精确的。回忆模拟信号的时域内插公式(3.33):

$$x_a(t)=\sum_{n=-\infty}^{\infty}x(n)\mathrm{sinc}[F_s(t-nT_s)] \tag{5.20}$$

DTFT 内插公式(5.19)与它很相象。

然而,也存在一些不同。第一,时域公式(5.20)重构一任意非周期模拟信号,而频域公式(5.19)给出一周期信号。第二,在(5.19)我们用$\frac{\sin(Nx)}{N\sin x}$内插函数代替更熟悉的$\frac{\sin x}{x}$(sinc)函数。因此 $\Phi(\omega)$函数有时也叫做数字 sinc 函数,它本身是一个周期函数。这是在例 5.2 中看到的函数。

MATLAB 实现

实际实现内插公式(5.19)时,它遇到与(5.20)同样的命运。为了由离散时间付利叶变换的样本 $\tilde{X}(k)$得到 $X(e^{j\omega})$,需要产生很多个内插多项式(5.18),并且计算它们的线性组合。更进一步说,用 MATLAB 实现时,我们需要在 $0\leqslant\omega\leqslant2\pi$ 区间上的小栅格上计算(5.19)。显然这不是一种有效的算法。另一种方法是采用三次样条内插函数作为对(5.19)的有效逼近。这正是在第三章中为实现(5.20)我们采用的方法。然而。存在另一种基于 DFT 的有效算法,我们将在下一节中研究。

离散付利叶变换

离散付利叶级数给我们提供了一种对离散时间付利叶变换作数值计算的技巧,同时它也

提出了时域出现混叠的潜在问题。数学分析表明对离散时间付利叶变换的采样,可得到一周期信号 $\tilde{x}(n)$。但在实际中大多数信号不具有周期性,它们很可能具有有限持续时间。对这些信号,怎样探讨一种可数值计算的付利叶表示式? 理论上,可通过定义一周期信号,其基本形状为有限持续时间信号,然后采用此周期信号的 DFS。实际上,我们定义一个新的变换,叫做离散付利叶变换(DFT),它是 DFS 的主周期。DFT 为对任意有限持续时间序列可数值计算的付利叶变换。

首先定义一个在区间 $0 \leqslant n \leqslant N-1$ 上具有 N 个样本点的有限持续时间序列 $x(n)$, $\tilde{x}(n)$ 是用 $x(n)$ 产生的周期为 N 的周期信号;由(5.19)得:

$$\tilde{x}(n)=\sum_{r=-\infty}^{\infty} x(n+rN)$$

这是一种比较繁琐的表示。采用对变量取模 N 的运算,将它简化为:

$$\tilde{x}(n)=x(n \bmod N) \tag{5.21}$$

下面对这种运算作一简单说明:若变量 n 为 0 和 $N-1$ 之间的一个数,那么运算后仍为它本身;否则从 n 中减或加上 N 的整数倍直至答数在 0 到 $N-1$ 之间。注意仅当 $x(n)$的长度等于或小于 N 时,(5.21)才有效。此外,我们用下面更简便的表示来表征模 N 运算。

$$x((n))_N \triangleq x(n \bmod N) \tag{5.22}$$

于是 $x(n)$和 $\tilde{x}(n)$之间的关系为:

$$\begin{aligned} \tilde{x}(n) &= x((n))_N && \text{(周期延伸)} \\ x(n) &= \tilde{x}(n) R_N(n) && \text{(加窗操作)} \end{aligned} \tag{5.23}$$

MATLAB 中的 rem(n,N)函数用来确定 n 除以 N 的余数。当 $n \geqslant 0$ 时,此函数可用来实现模 N 运算。$n<0$ 时,为得到正确值,需要对结果进行修正。这由下面的 m = mod(n ,N)函数完成。

```
function m = mod(n,N)
% 计算 m = (n mod N) 下标
% ——————————————
% m = mod(n,N)
m = rem(n,N);
m = m + N;
m = rem(m,N);
```

在此函数中,n 可以是任意整数数组,数组 m 返回 n 的模 N 值。

从频率采样定理知道，N 点序列 $x(n)$ 的 N 个离散时间付利叶变换 $X(e^{j\omega})$ 等间隔样本能唯一地重构 $X(e^{j\omega})$。这些单位圆上的 N 个样本叫做离散付利叶系数。设 $\tilde{X}(k)=\text{DFS}\tilde{x}(n)$，为一周期(具有无限持续时间)序列，则它的主周期为具有有限持续时间的离散付利叶变换。下面的定义清楚地表明这一点。N 点序列的离散付利叶变换由下式给出：

$$X(k)\triangleq\text{DFT}[x(n)]=\begin{cases}\tilde{X}(k), & 0\le k\le N-1\\ 0, & \text{其它}\end{cases}=\tilde{X}(k)R_N(k)$$

或者

$$X(k)=\sum_{n=0}^{N-1}x(n)W_N^{nk},\ 0\le k\le N-1 \tag{5.24}$$

注意 DFT $X(k)$ 也为 N 点序列，就是说，它在 $0\leqslant k\leqslant N-1$ 外无定义。由(5.23)，$X(k)=X((k))_N$；即在 $0\leqslant k\leqslant N-1$ 外，仅 DFS$\tilde{X}(k)$ 有定义，当然它是 $X(k)$ 的周期延伸。最后，$X(k)=\tilde{X}(k)R_N(k)$ 表明 DFTX(k) 是 $\tilde{X}(k)$ 的主值周期。

N 点 DFT $X(k)$ 的离散付利叶反变换为：

$$x(n)\triangleq\text{IDFT}[X(k)]=\tilde{x}(n)R_N(n)$$

或者

$$x(n)=\frac{1}{N}\sum_{k=0}^{N-1}X(k)W_N^{-kn},\ 0\leqslant n\leqslant N-1 \tag{5.25}$$

同样 $x(n)$ 在 $0\leqslant n\leqslant N-1$ 外无定义，它在此范围外的延伸为 $\tilde{x}(n)$。

MATLAB 实现

由上面的讨论可知，在 $0\leqslant n\leqslant N-1$ 上，DFT 与 DFS 相同。因此可用类似的方法实现 DFT。若以列向量 $\mathbf{x}$，$\mathbf{X}$ 形式排列 $x(n)$ 和 $X(k)$，由(5.24)、(5.25)得到：

$$\begin{aligned}\mathbf{X}&=\mathbf{W}_N\mathbf{x}\\ \mathbf{x}&=\frac{1}{N}\mathbf{W}_N^*\mathbf{X}\end{aligned} \tag{5.26}$$

其中 $\mathbf{W}_N$ 为(5.7)中定义的矩阵，现在称为 DFT 矩阵。原先名为 dfs 和 idfs 的 MATLAB 函数各改名为 dft 和 idft 函数，来实现离散付利叶变换运算。

```
function [Xk] = dft(xn,N)
```

```
% 计算离散付利叶变换
% ——————————
% [Xk]=dft(xn,N)
% Xk=在0 <= n <= N-1间的DFT 系数数组
% xn=N点有限持续时间序列
% N=DFT的长度
%
n=[0:1:N-1];                % n的行向量
k=[0:1:N-1];                % k的行向量
WN=exp(-j*2*pi/N);          % Wn因子
nk=n'*k;                    % 产生一个含nk值的N乘N维矩阵
WNnk=WN .^ nk;              % DFT 矩阵
Xk=xn * WNnk;               % DFT 系数的行向量

function [xn]=idft(Xk,N)
% 计算逆离散付利叶变换
% ——————————
% [xn]=idft(Xk,N)
% xn=在0 <= n <= N-1间的N点有限持续时间序列
% Xk=在0 <= k <= N-1间的DFT 系数数组
% N=DFT的长度
%
n=[0:1:N-1];                % n的行向量
k=[0:1:N-1];                % k的行向量
WN=exp(-j*2*pi/N);          % Wn因子
nk=n'*k;                    % 产生一个含nk值的N乘N维矩阵
WNnk=WN .^(-nk);            % IDFT 矩阵
xn=(Xk * WNnk)/N;           % IDFT 的行向量
```

例5.6 $x(n)$是一4点序列:

$$x(n)=\begin{cases}1, & 0\le n\le 3\\ 0, & 其它\end{cases}$$

a. 计算离散时间付利叶变换 $X(e^{j\omega})$,并且画出它的幅度和相位。

b. 计算 $x(n)$的4点DFT。

解: a. 离散时间付利叶变换为:

$$X(e^{j\omega})=\sum_{0}^{3}x(n)e^{-j\omega n}=1+e^{-j\omega}+e^{-j2\omega}+e^{-j3\omega}$$

$$=\frac{1-e^{-j4\omega}}{1-e^{-j\omega}}=\frac{\sin(2\omega)}{\sin(\omega/2)}e^{-j3\omega/2}$$

因而

$$|X(e^{j\omega})|=\left|\frac{\sin(2\omega)}{\sin(\omega/2)}\right|$$

和

$$\angle X(e^{j\omega})=\begin{cases}-\frac{3\omega}{2}, & 当\frac{\sin(2\omega)}{\sin(\omega/2)}>0\\ -\frac{3\omega}{2}\pm\pi, & 当\frac{\sin(2\omega)}{\sin(\omega/2)}<0\end{cases}$$

这些图如图 5.4 所示。

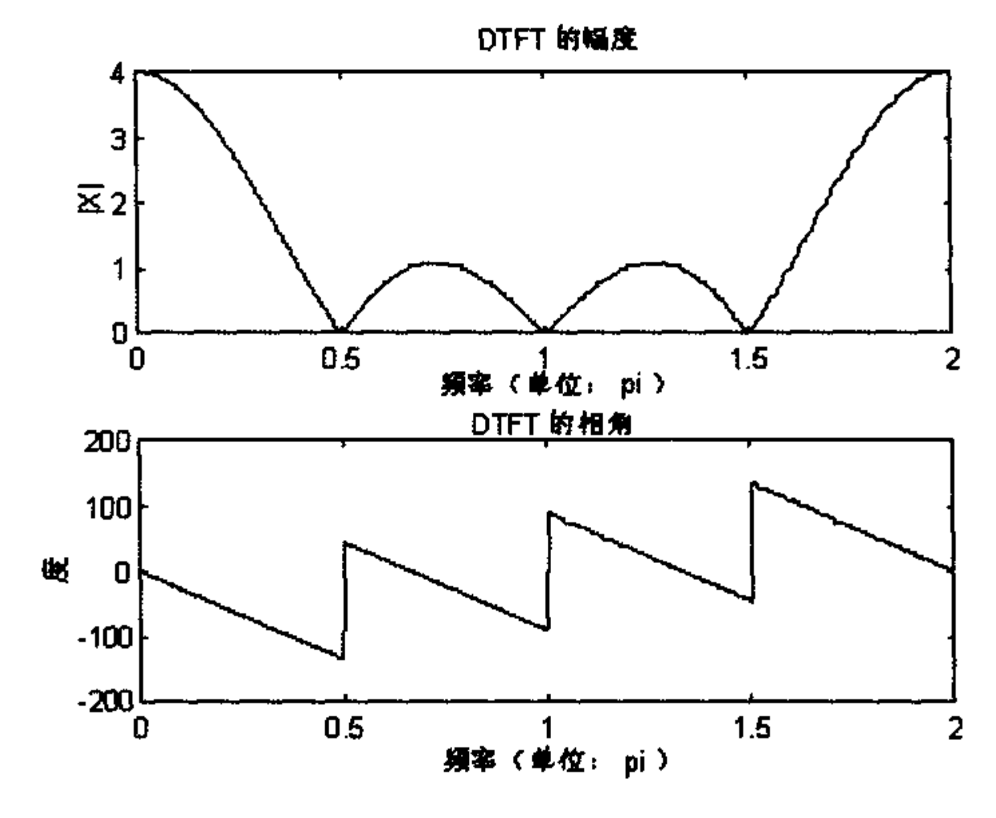


图 5.4　例 5.6 中的 DTFT 图

b. 用 $X_4(k)$ 表示 4 点 DFT：

$$X_4(k)=\sum_{n=0}^{3}x(n)W_4^{nk};k=0,1,2,3;W_4=e^{-j2\pi/4}=-j$$

这些计算与例 5.1 类似。我们也可用 MATLAB 计算 DFT。

```
>> x=[1,1,1,1]; N=4;
>> X=dft(x,N);
magX=abs(X), phaX=angle(X)*180/pi
magX =
4.0000    0.0000    0.0000    0.0000
phaX =
0  -134.9822  -90.0000  -44.9979
```

因而

$$X_4(k)=\{\underset{\uparrow}{4},0,0,0\}$$

注意当幅度样本为零时,相应的相角不为零。这是由于用 MATLAB 计算相角的特定算法造成的。一般来讲,这些角度可以忽略。DFT 的图如图 5.5 所示。从图看出,$X_4(k)$正确给出了 $X(e^{j\omega})$的 4 个样本,但它只有一个非零样本。这令人惊奇吗?考察一个全 1 的 4 点 $x(n)$,它的周期性延伸为

$$\tilde{x}(n)=1,\forall n$$

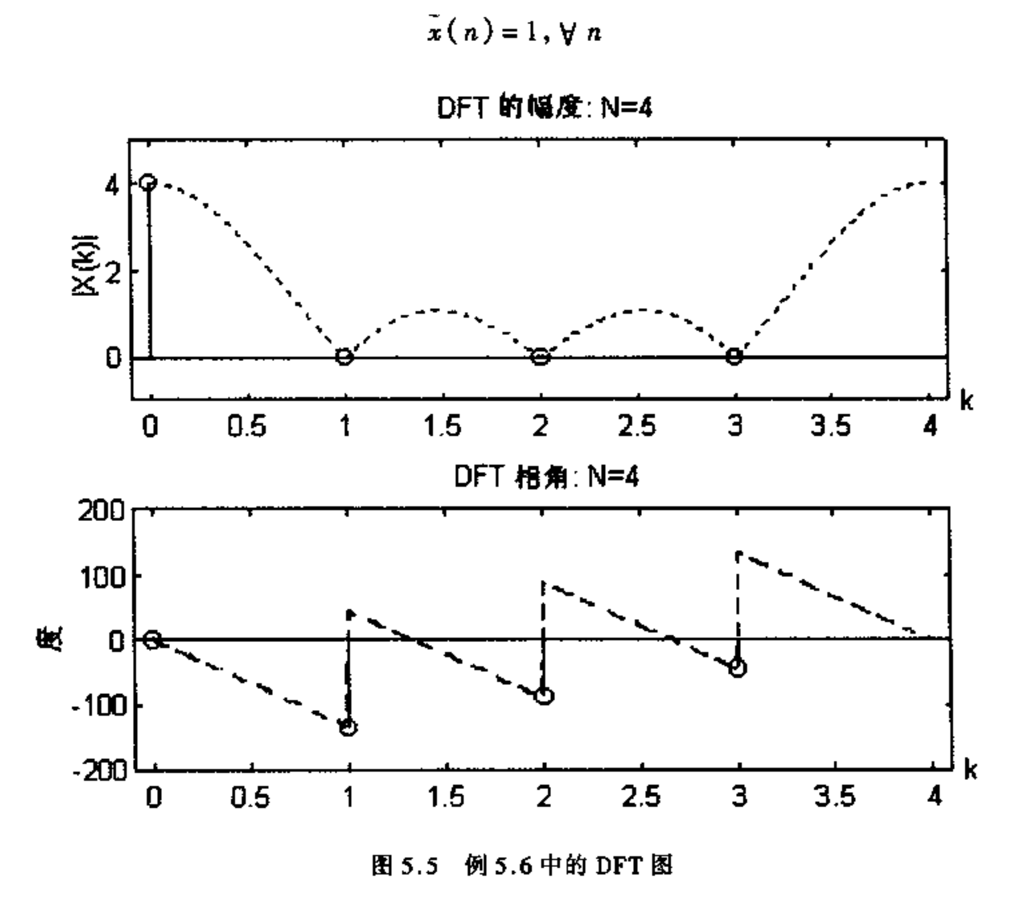


图 5.5　例 5.6 中的 DFT 图

它是一恒定(或 DC)信号。这正是由 DFT $X_4(k)$推知的,它在 $k=0$(或 $\omega=0$)上有一非零样本,而在其他频率上为零。

例 5.7 怎样得到 DTFT $X(e^{j\omega})$的其他样本?

解: 显然,我们的采样频率应更小一些;也就是说,我们应增加 N。假定将点数增加一倍,或 $N=8$ 而不是 4。我们可给 $x(n)$后附上 4 个零得到一 8 点序列。

$$x(n)=\{\underset{\uparrow}{1},1,1,1,0,0,0,0\}$$

这是一个很重要的运算,叫做填零运算。在实际应用中,为了得到一较密的频谱,这种运算是很必要的。设 $X_8(k)$为一 8 点 DFT,则

$$X_8(k)=\sum_{n=0}^{7}x(n)W_8^{nk},\ k=0,1,\ldots,7;\ W_8=e^{-j\pi/4}$$

在这种情况下,频率分辨率为 $\omega_1=2\pi/8=\pi/4$。

```
>> x=[1,1,1,1,zeros(1,4)]; N=8;
>> X=dft(x,N);
>> magX=abs(X), phaX=angle(X)*180/pi
magX =
   4.0000  2.6131  0.0000  1.0824  0.0000  1.0824  0.0000    2.6131
phaX =
   0  -67.5000   -134.9822   -22.5000   -90.0000  22.5000   -44.9979  67.
5000
```

从而

$$X_8(k)=\{\underset{\uparrow}{4},2.6131e^{-j67.5°},0,1.0824e^{-j22.5°},0,1.0824e^{j22.5°},0,2.6131e^{j67.5°}\}$$

如图 5.6 所示。更进一步,我们给 $x(n)$填充 12 个零,成为一 16 点序列,即

$$x(n)=\{\underset{\uparrow}{1},1,1,1,0,0,0,0,0,0,0,0,0,0,0,0\}$$

那么其频率分辨率为 $\omega_1=2\pi/16=\pi/8$,$W_{16}=e^{-j\pi/8}$。因此我们得到频谱样本间隔为 $\pi/8$ 的更密的频谱。$X_{16}(k)$如图 5.7 所示。

结论: 基于以上两个例子,可得出以下结论。

1. 填零是给原始序列填零的运算。这导致较长的 DFT,它会给原始序列的离散时间付利叶变换提供间隔较近的样本。在 MATLAB 中,用 zeros 函数实现填零运算。

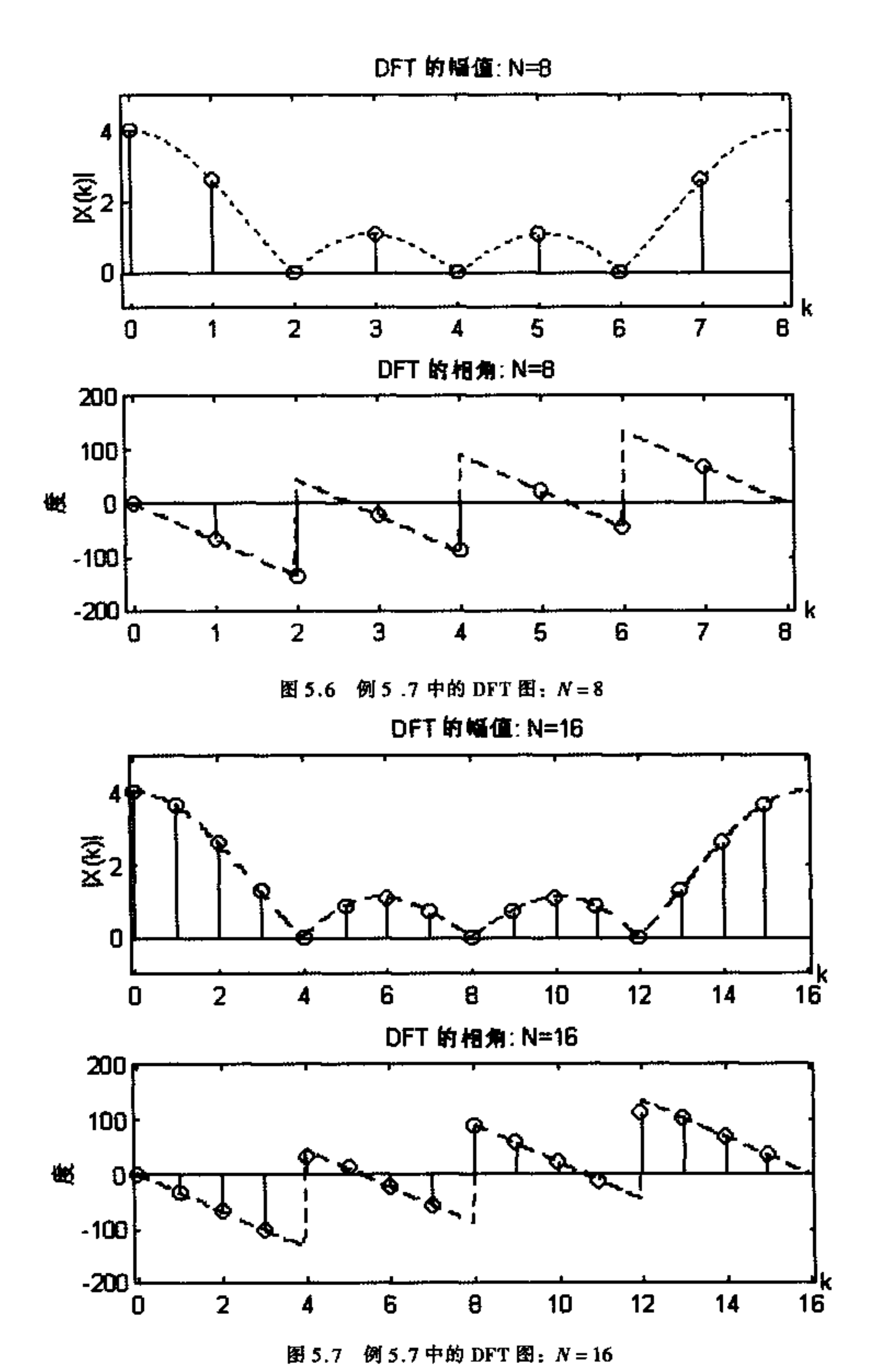


图 5.6 例 5 .7 中的 DFT 图：$N=8$

图 5.7 例 5.7 中的 DFT 图：$N=16$

2．例 5.6 中，为精确画出 $x(n)$ 的离散时间付利叶变换 $X(e^{j\omega})$，只需要 4 点 DFT $X_4(k)$。这是因为 $x(n)$ 仅有 4 个非零样本，因此可对 $X_4(k)$ 用内插公式(5.19)得到 $X(e^{j\omega})$。然而，实

际上,更简便的方法是得到 $X_8(k)$, $X_{16}(k)$等等,用它们来填充 $X(e^{j\omega})$的值,而不是用内插公式。若用快速付利叶变换算法计算 DFT,这种方法会更有效。

3. 填零运算给我们提供了一个较密的频谱和较好的图示形式。但因为在信号中只是附加了零,而没有增加任何新的信息,因此它不能提供高分辨率的频谱。

4. 为得到高分辨率的频谱,人们得从实验或观察中取得更多的数据(如例 5.8 所示)。其他的先进办法则是利用边缘信息和非线性技术。

例 5.8 为了说明高密度频谱与高分辨率频谱之间的区别,考察序列

$$x(n) = \cos(0.48\pi n) + \cos(0.52\pi n)$$

我们想求出它基于有限个样本的频谱。

a. 当 $0 \leq n \leq 10$ 时,确定并画出 $x(n)$的离散时间付利叶变换。

b. 当 $0 \leq n \leq 100$ 时,确定并画出 $x(n)$的离散时间付利叶变换。

解:我们可解析地求出其离散时间付利叶变换,但 MATLAB 是研究这些问题的有力工具。

a. 首先确定 $x(n)$的 10 点 DFT,得到其离散时间付利叶变换的估计

```
>> n = [0:1:99]; x = cos(0.48 * pi * n) + cos(0.52 * pi * n);
>> n1 = [0:1:9]; y1 = x(1:1:10);
>> subplot(2,1,1); stem(n1,y1); title('信号 x(n), 0 < = n < = 9');
>> axis([0,10, -2.5,2.5]); text(10.2, -2.5,'n')
>> Y1 = dft(y1,10); magY1 = abs(Y1(1:1:6));
>> k1 = 0:1:5; w1 = 2 * pi/10 * k1;
>> subplot(2,1,2); plot(w1/pi,magY1); title('DTFT 幅度的样本');
>> xlabel('频率(单位:pi)')
```

图 5.8 表明,没有足够的样本数来下结论。因此我们填 90 个零以得到密频谱。

```
>> n2 = [0:1:99]; y2 = [x(1:1:10) zeros(1,90)];
>> subplot(2,1,1); stem(n2,y2); title('信号 x(n), 0 < = n < = 9 + 90 zeros');
>> Y2 = dft(y2,100); magY2 = abs(Y2(1:1:51));
>> k2 = 0:1:50; w2 = 2 * pi/100 * k2;
>> subplot(2,1,2); plot(w3/pi,magY3); title('DTFT 幅度');
>> xlabel('频率(单位:pi)')
```

图 5.9 表明此序列在 $\omega = 0.5\pi$ 处有一主频率,原始序列则没有说明这一点,它有两个频率。填零运算提供了如图 5.8 所示的较平滑的频谱。

b. 为得到更好的频谱信息,我们将用 $x(n)$的头 100 个样本来确定它的离散时间付利叶变换。

```
>> subplot(2,1,1); stem(n,x);
```

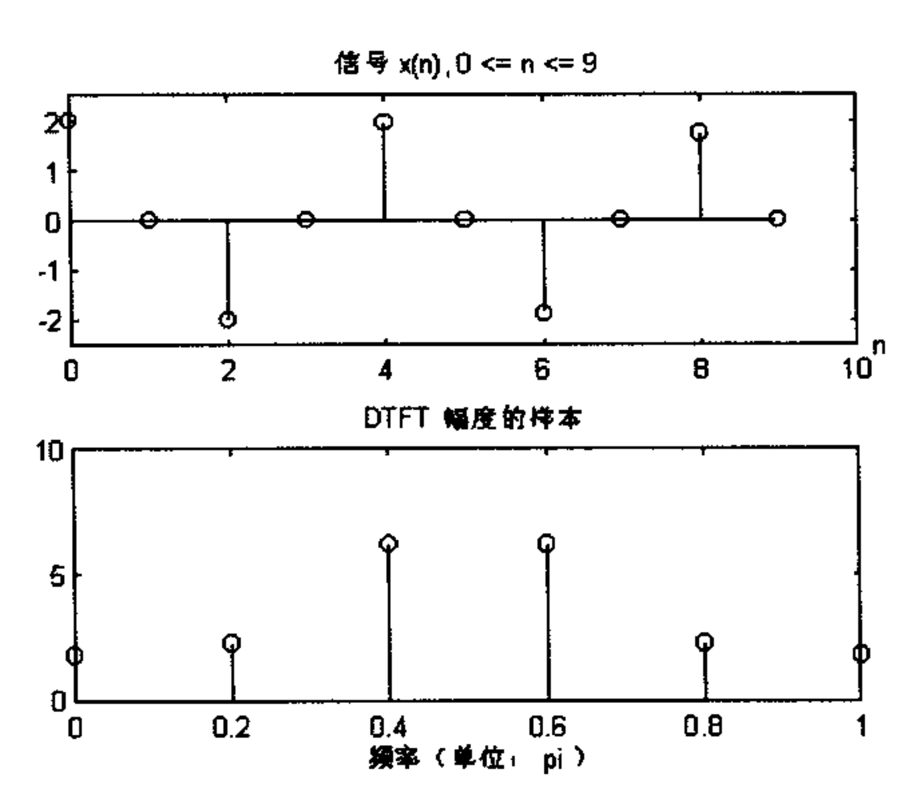


图 5.8　例 5.8a 中的信号与频谱 $N=10$

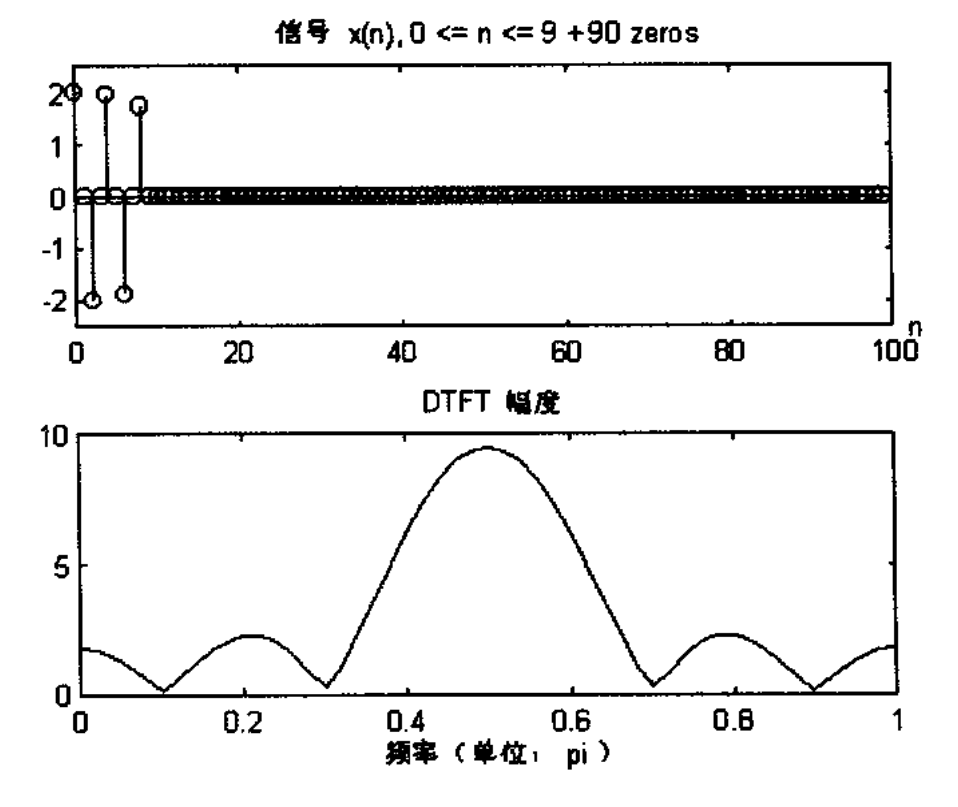


图 5.9　例 5.8a 中的信号与频谱 $N=100$

```
>> title('信号 x(n),0<=n<=99');xlabel('n')
>> X=dft(x,100);magX=abs(X(1:1:51));
>> k=0:1:50;w=2*pi/100*k;
>> subplot(2,1,2);plot(w/pi,magX);title('DTFT 幅度');
>> xlabel('频率(单位:pi)')
```

图 5.10 中的离散付利叶变换图清楚地表明两个靠得很近的频率。这是 $x(n)$ 的高分辨率频谱。给序列尾部填零至 100 个点,将会导致一更光滑的频谱,但不能提供更多的新信息。我们鼓励学生自己来证明这一点。

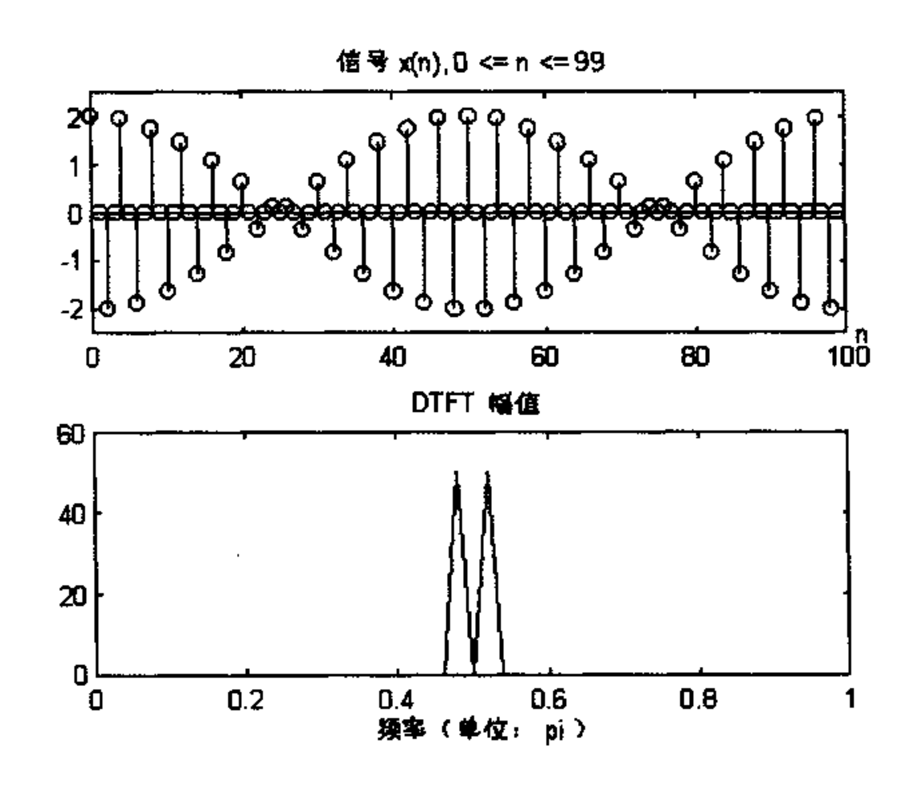


图 5.10　例 5.8b 中的信号与频谱 $N=100$

离散付利叶变换的性质

由于 DFS 是数学上有根据的表示式,因此 DFT 性质可由 DFS 的性质导出。我们将不加证明地给出一些有用的结论。这些性质经必要的修改也适用于 DFS。设 $X(k)$ 是序列 $x(n)$ 的 N 点 DFT。除非特别声明,这些性质均采用 N 点 DFT。

1.线性性质:DFT 是一种线性变换

$$\mathrm{DFT}[ax_1(n)+bx_2(n)]=a\mathrm{DFT}[x_1(n)]+b\mathrm{DFT}[x_2(n)] \tag{5.27}$$

注意:若 $x_1(n)$ 和 $x_2(n)$ 的长度不同,也就是说,它们分别是 N_1 点和 N_2 点序列,那么选择 N_3

$=\max(N_1,N_2)$，将它们作 N_3 点 DFT 处理。

2.循环折叠性质：若把 N 点序列折叠。那么结果 $X(-n)$将不再是一个 $0\leqslant n\leqslant N-1$ 上的 N 点序列，也不可能计算它的 DFT。因此对变量$(-n)$进行取模 N 运算，定义折叠为：

$$x((-n)_N=\begin{cases}x(0), & n=0\\ x(N-n), & 1\leq n\leq N-1\end{cases} \tag{5.28}$$

这叫做循环折叠。为了更形象化一些，想象序列 $x(n)$以反时针方向放置在一个圆上，使下标为 $n=0$ 和 $n=N$ 的两点重叠。故 $x((-n))N$ 可看成 $X(n)$沿着圆的顺时针放置。在 MATLAB 中，循环折叠由 x = x(mod(-n,N)+1)得到。注意 MATLAB 中的变量下标由 1 开始，因此它的 DFT 由下式给出：

$$\text{DFT}[x((-n))_N]=X((-k))_N=\begin{cases}X(0), & k=0\\ X(N-k), & 1\leqslant k\leq N-1\end{cases} \tag{5.29}$$

例 5.9 设 $x(n)=10(0.8)n$ ，$0\leqslant n\leqslant 10$

a. 确定并画出 $x((-n))11$

b. 验证循环折叠性质

解：a. MATLAB 实现

```
>> n=0:100; x=10*(0.8) .^ n;
>> y=x(mod(-n,11)+1);
>> subplot(2,1,1); stem(n,x); title('初始序列')
>> xlabel('n');ylabel('x(n)');
>> subplot(2,1,2); stem(n,y); title('循环折叠序列')
>> xlabel('n');ylabel('x(-n mod 11)');
```

图 5.11 给出循环折叠的结果。

b. MATLAB 实现

```
>> X=dft(x,11); Y=dft(y,11);
>> subplot(2,2,1); stem(n,real(X));
>> title('|DFT[x(n)]|的实部');xlabel('k');
>> subplot(2,2,2); stem(n,imag(X));
>> title('|DFT[x(n)]|的虚部');xlabel('k');
>> subplot(2,2,3); stem(n,real(Y));
>> title('|DFT[x((-n))11]|的实部');xlabel('k');
>> subplot(2,2,4); stem(n,imag(Y));
>> title('|DFT[x((-n))11]|的虚部');xlabel('k');
```

图 5.12 则验证了此性质。

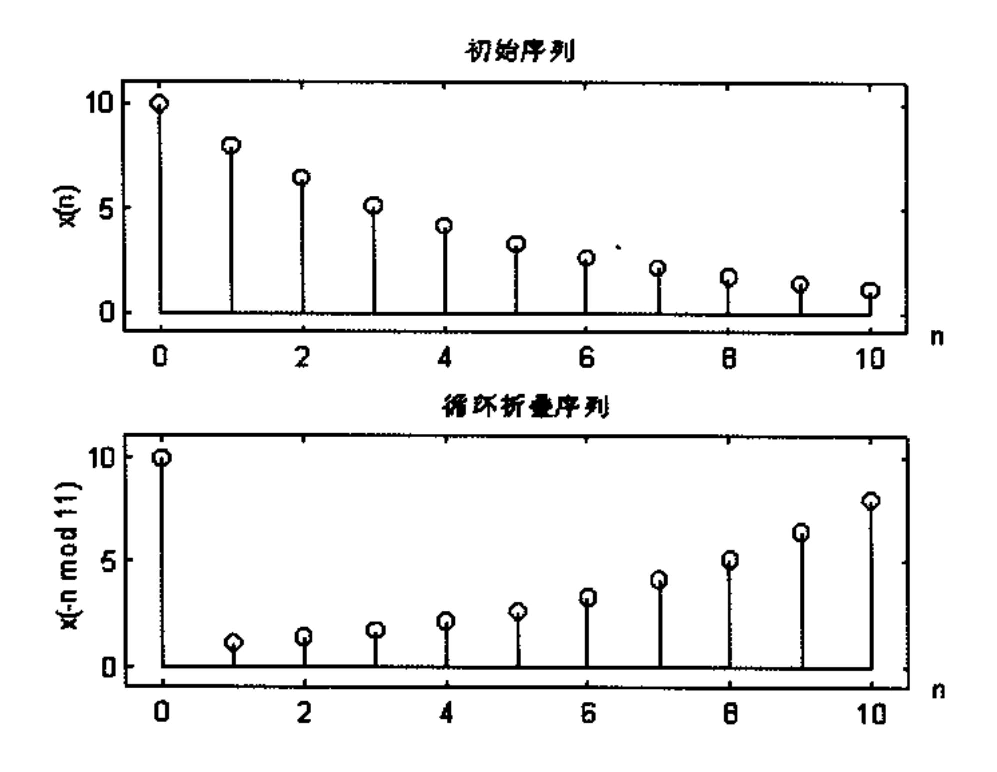


图 5.11 例 5.9a 中的循环折叠性质

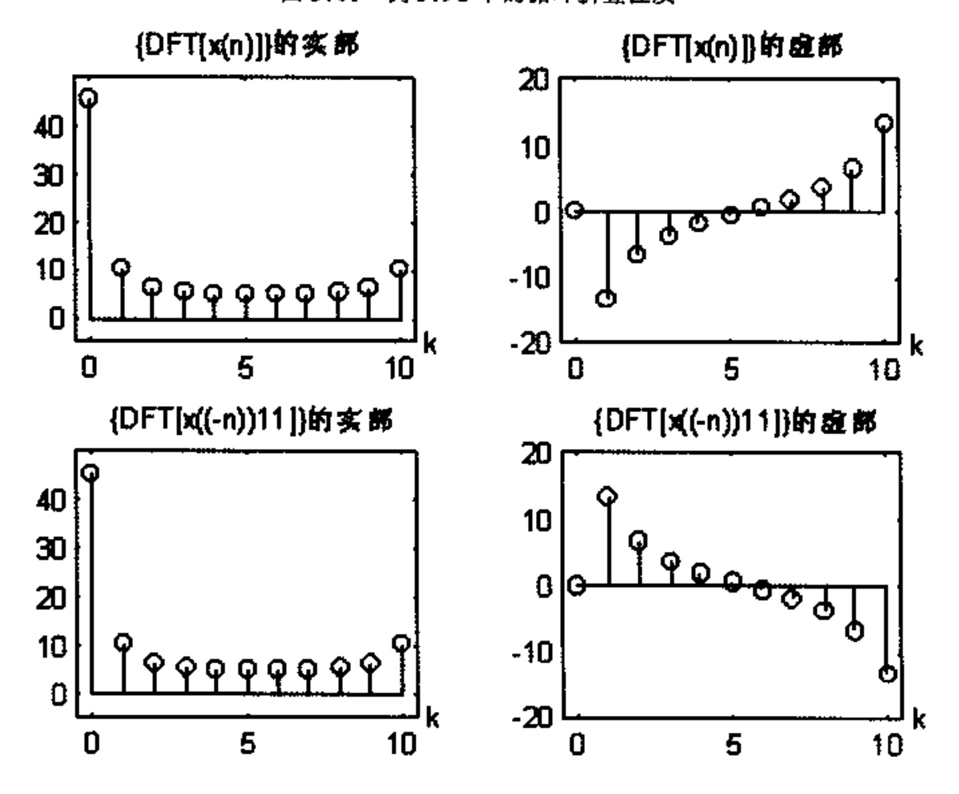


图 5.12 例 5.9b 中的循环折叠性质

3. 共轭性质:与上性质类似,我们将引人频域中的循环折叠:

$$\text{DFT}[x^*(n)] = X^*((-k))_N \tag{5.30}$$

4. 实序列的对称性质:设 $x(n)$为 N 点实序列,则 $x(n)=x^*(n)$,利用上面的性质,得:

$$X(k) = X^*((-k))_N \tag{5.31}$$

此对称性叫做循环共轭对称。进一步地,可得:

$$\left.\begin{array}{ll}
\text{Re}|X(k)| = \text{Re}|X((-k))_N| & \Rightarrow \text{循环偶序列} \\
\text{Im}|X(k)| = -\text{Im}|X((N-k))_N| & \Rightarrow \text{循环奇序列} \\
|X(k)| = |X((-k))_N| & \Rightarrow \text{循环偶序列} \\
\angle X(k) = -\angle X((-k))_N & \Rightarrow \text{循环奇序列}
\end{array}\right\} \tag{5.32}$$

讨论:1. 观察例 5.6 和 5.7 中各种 DFT 的幅度和相位可知,它们确实满足上述的循环对称性质。这些对称与一般的奇偶对称不同,为了更形象化一些,想象 DFT 样本沿一个圆放置,使下标为 $n=0$ 和 $n=N$ 的点重叠。因而样本可看作关于 $k=0$ 对称,这与它的名字折叠对称相符。

2. 相应地,DFS 系数的对称性叫做周期性共轭对称。

3. 既然 DFT 具有对称性,为得到 $X(k)$,只需计算

$$k=0,1,\ldots,\frac{N}{2};\quad N \text{ 为偶数时}$$

或 $$k=0,1,\ldots,(N-1)/2;\ N \text{ 为奇数时}$$

这样就会节约 50%的计算量和内存。

4. 从(5.30)可知

$$X(0) = X^*((-0))_N = X^*(0)$$

这说明当 $k=0$ 时的 DFT 系数肯定是一个实数。$k=0$ 意味着 $\omega_k = k\omega_1 = 0$ 为一直流频率,实 $X(n)$的直流分量肯定是实数。另外,若 N 为偶数,则 $N/2$ 也是一整数。由(5.32)得

$$X\left(\frac{N}{2}\right) = X^*\left(\left(-\frac{N}{2}\right)\right)_N = X^*\left(\frac{N}{2}\right)$$

这表明甚至当 $k=N/2$ 时的分量也为实数。由于 $k=N/2$ 意味着频率 $\omega_{N/2}=(N/2)(2\pi/N)=\pi$,即为数字奈奎斯特(Nyquist)频率,所以此分量称为奈奎斯特分量。

如第二章中讨论的那样,实信号也能分解成它的奇、偶分量 $x_e(n)$和 $x_o(n)$。然而这些分

量不再是一个 $0\leqslant n\leqslant N-1$ 上的 N 点序列，因此不能计算它们的 N 点 DFT。我们用上面讨论的循环折叠定义一套新的分量，这些循环偶分量与循环奇分量的定义为：

$$
\begin{aligned}
x_{ec}(n) &\triangleq \frac{1}{2}[x(n)+x((-n))_N]=\begin{cases}x(0), & n=0\\ \frac{1}{2}[x(n)+x(N-n)], & 1\leq n\leq N-1\end{cases}\\
x_{oc}(n) &\triangleq \frac{1}{2}[x(n)-x((-n))_N]=\begin{cases}0, & n=0\\ \frac{1}{2}[x(n)-x(N-n)], & 1\leq n\leq N-1\end{cases}
\end{aligned}
\tag{5.33}
$$

因而

$$
\begin{aligned}
\text{DFT}[x_{ec}(n)] &= \text{Re}[X(k)]=\text{Re}[X((-k))_N]\\
\text{DFT}[x_{oc}(n)] &= \text{Im}[X(k)]=\text{Im}[X((-k))_N]
\end{aligned}
\tag{5.34}
$$

隐含意义：若 $x(n)$ 为实的循环偶序列，那么它的 DFT 也是实的循环偶序列，因此只计算头 $0\leqslant n\leqslant N/2$ 个系数就可得到它的 DFT。

利用(5.33)，可很简单地探讨出把 N 点序列分解成它的循环奇、偶分量。下面的 circevod 函数采用了早先给出的 mod 函数实现 n 取模 N 运算。

```
function [xec, xoc] = circevod(x)
% 将信号分解为循环偶和循环奇两部分
% ———————————————————————————
% [xec, xoc] = circecod(x)
%
if any(imag(x) ~= 0)
    error('x 非实数序列')
end
N = length(x); n = 0:(N-1);
xec = 0.5 * (x + x(mod(-n,N)+1));
xoc = 0.5 * (x - x(mod(-n,N)+1));
```

例 5.10 与例 5.9 相同，设 $x(n)=10(0.8)^n, 0\leqslant n\leqslant 10$

a. 分解并画出 $x(n)$ 的 $x_{ec}(n)$ 和 $x_{oc}(n)$ 分量。

b. 验证(5.34)中的性质。

解：a. MATLAB 实现

```
>> n = 0:10; x = 10 * (0.8) .^ n;
>> [xec,xoc] = circevod(x);
>> subplot(2,1,1); stem(n,xec); title('循环偶分量')
```

```
>> xlabel('n');ylabel('xec(n)'); axis([-0.5,10.5,-1,11]);
>> subplot(2,1,2); stem(n,xoc); title('循环奇分量')
>> xlabel('n');ylabel('xoc(n)');axis([-0.5,10.5,-4,4]);
```

图 5.13 给出 $x(n)$的循环对称分量。

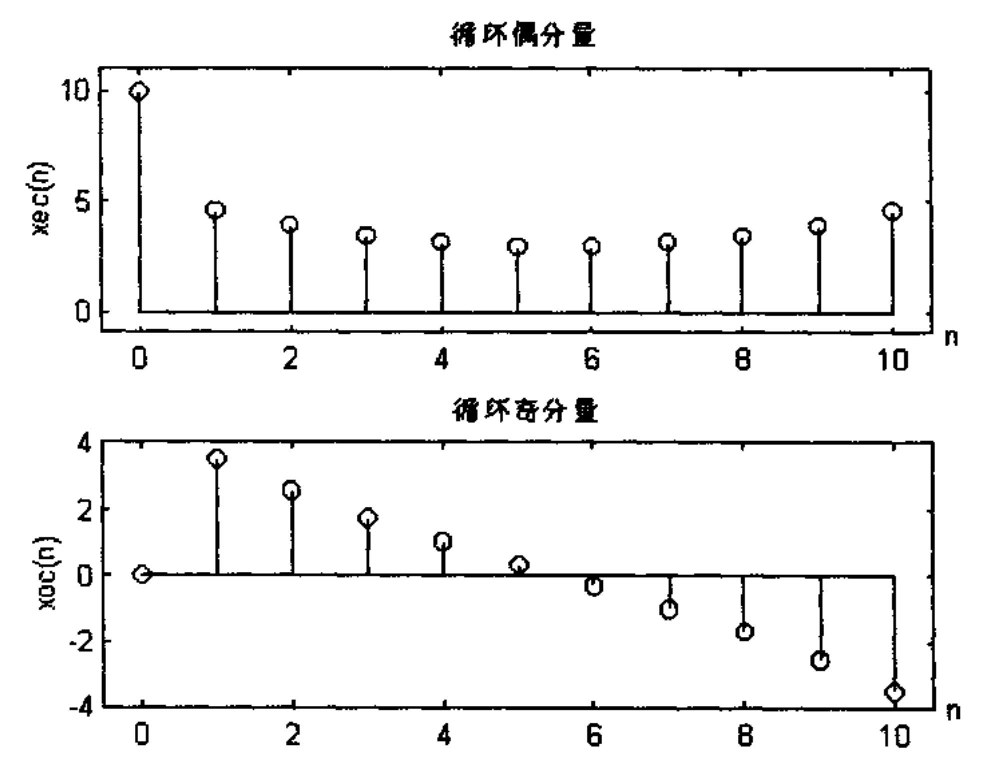


图 5.13 例 5.10a 中的循环偶分量和循环奇分量

b. MATLAB 实现

```
>> X=dft(x,11); Xec=dft(xec,11); Xoc=dft(xoc,11);
>> subplot(2,2,1); stem(n,real(X)); axis([-0.5,10.5,-5,50])
>> title('|DFT[x(n)]|的实部');xlabel('k');
>> subplot(2,2,2); stem(n,imag(X)); axis([-0.5,10.5,-20,20])
>> title('|DFT[x(n)]|的虚部');xlabel('k');
>> subplot(2,2,3); stem(n,real(Xec)); axis([-0.5,10.5,-5,50])
>> title('DFT[xec(n)]');xlabel('k');
>> subplot(2,2,4); stem(n,imag(Xoc)); axis([-0.5,10.5,-20,20])
>> title('DFT[xoc(n)]');xlabel('k');
```

观察图 5.14 中的各图,可知 $x_{ec}(n)$的 DFT 与 $X(k)$的实部相同,$x_{oc}(n)$的 DFT 与 $X(k)$的虚部相同。

复序列的类似性质我们将在习题 5.10 中研究。

5.序列的循环移位: 如果 N 点序列沿一方向移位,它将不再位于区间 $0\leqslant n\leqslant N-1$ 上。

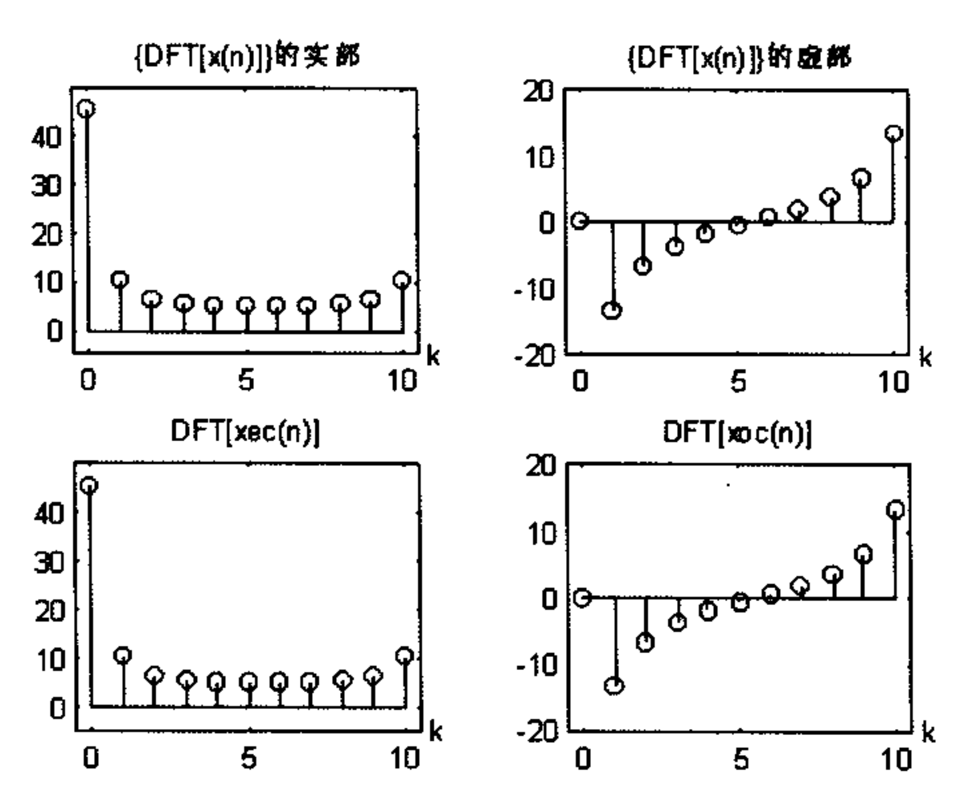


图 5.14 例 5.10b 中的 DFT 对称性质

我们首先将 $x(n)$延伸为它的周期序列 $\tilde{x}(n)$,然后移位 m 个样本,得到:

$$\tilde{x}(n-m)=x((x-m))_N \tag{5.35}$$

这叫做 $\tilde{x}(n)$的周期移位,把移位后的序列转换成一 N 点序列,所得序列为:

$$\tilde{x}(n-m)R_N(n)=x((n-m))_N R_N(n) \tag{5.36}$$

叫做 $x(n)$的循环移位。为了形象化,再次想象把 $x(n)$沿一圆固定放置,将此圆旋转 k 个样本,在 $0\leqslant n\leqslant N-1$ 上展开此序列。它的 DFT 由下式给出:

$$\mathrm{DFT}[x((n-m))_N R_N(n)]=W_N^{km}X(k) \tag{5.37}$$

例 5.11 设 $x(n)=10(0.8)^n,0\leqslant n\leqslant 10$ 为 11 点序列

a. 画出 $x((n+4))_{11}R_{11}(\mathrm{n})$,也就是向左循环移位 4 个样本的序列;

b. 画出 $x((n-3))_{15}R_{15}(n)$,也就是说,假定 $x(n)$为 15 点序列,向右循环移位 3 个样本;

解:我们将用逐步画图的方法说明循环移位运算。此方法首先给出 $x(n)$的周期延伸序列 $\tilde{x}(n)=x((n))_N$,随后对 $\tilde{x}(n)$进行线性移位,得到 $\tilde{x}(n-m)=x((n-m))_N$,最后截取 $\tilde{x}(n-m)$得到循环移位。

a. 图 5.15 给出 4 个序列,左上角为 $x(n)$,左下角为 $\tilde{x}(n)$,右上角为 $\tilde{x}(n+4)$,右下角为 $x((n+4))_{11}R_{11}(n)$。特别注意当样本从一个方向移出$[0,N-1]$,它们将从相反方向再现。这就是循环意味的含义,它不同于线性移位。

b. 这种情形下,给 $x(n)$后填 4 个零,将其看作一个 15 点序列。此时的循环移位与 $N=11$ 时不同,如图 5.16 所示。事实上,循环移位 $x((n-3))_{15}$看起来像个线性移位 $x(n-3)$。

为实现循环移位,我们毋须像例 5.11 一样先得到周期移位。可用两种方法直接实现,第一种方法,在时域内对变量$(n-m)$实行取模 N 运算。这由下面的 cirshftt 函数实现。

```
function y = cirshftt(x,m,N)
% 长度为 N 的 x 序列:(时域)作 m 个样本的循环移位
% ------------------------------------------
% [y] = cirshftt(x,m,N)
% y = 包含循环移位的输出序列
% x = 长度 <= N 的输入序列
% m = 移位样本数
% N = 循环缓冲器长度
% 方法: y(n) = x((n-m) mod N)

% 检查 x 的长度
if length(x) > N
      error('N 必须 >= x 的长度')
end
x = [x zeros(1,N-length(x))];
n = [0:1:N-1];
n = mod(n-m,N);
y = x(n+1);
```

第二种方法,可在频域中用性质 5.37,这在习题 5.12 中探讨。

例 5.12 给定 11 点序列 $x(n)=10(0.8)^n, 0\leq n\leq 10$,求出并画出 $x((n-6))_{15}$

解:MATLAB 程序

```
>> n = 0:10; x = 10 * (0.8) .^ n;
>> y = cirshftt(x,6,15);
>> n = 0:14; x = [x, zeros(1,4)];
>> subplot(2,1,1); stem(n,x); title('初始序列')
>> xlabel('n');ylabel('x(n)');
>> subplot(2,1,2); stem(n,y);
>> title('循环移位序列, N = 15')
>> xlabel('n');ylabel('x((n-6) mod 15)');
```

结果如图 5.17 所示。

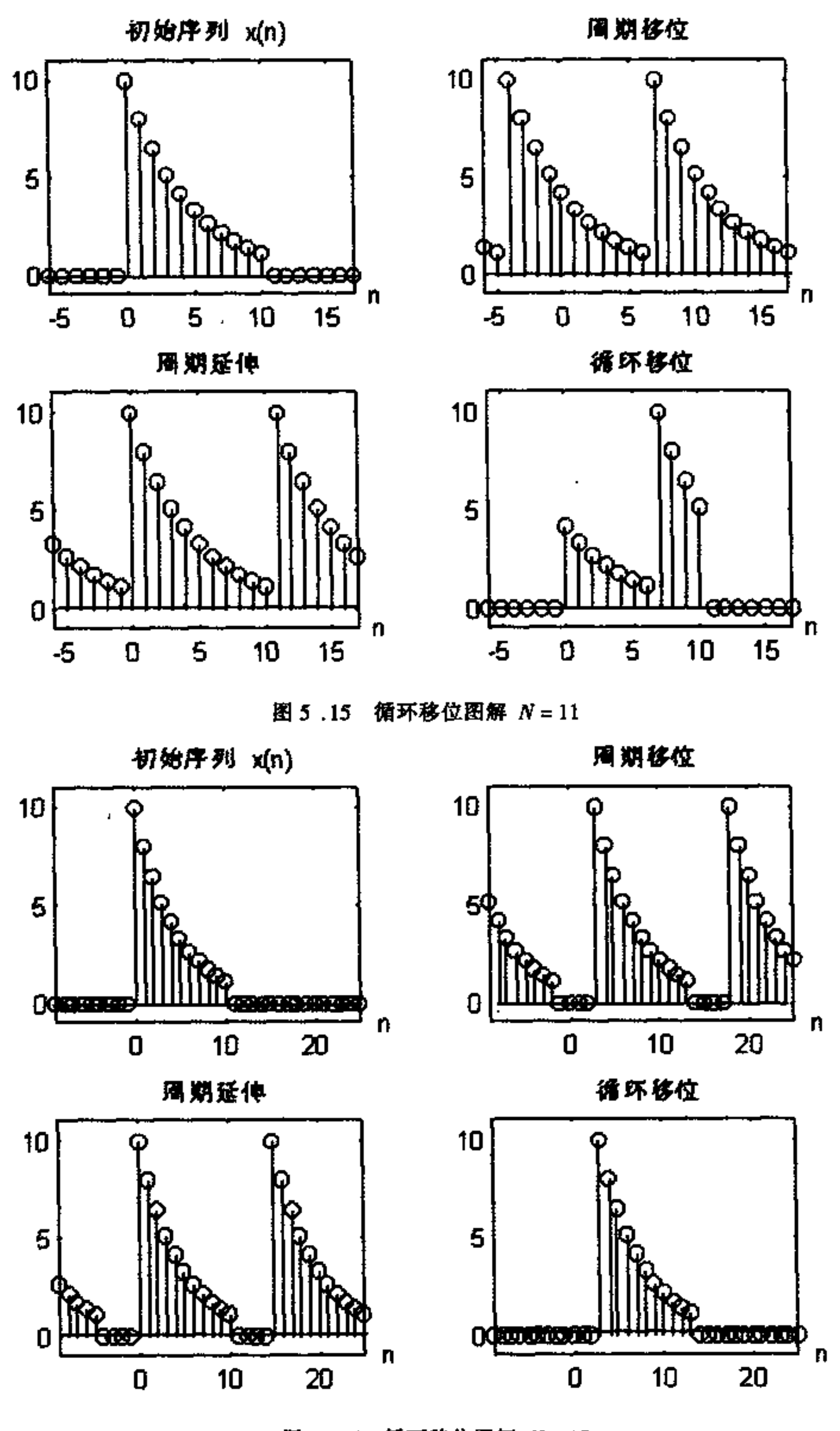


图 5.15 循环移位图解 $N = 11$

图 5.16 循环移位图解 $N = 15$

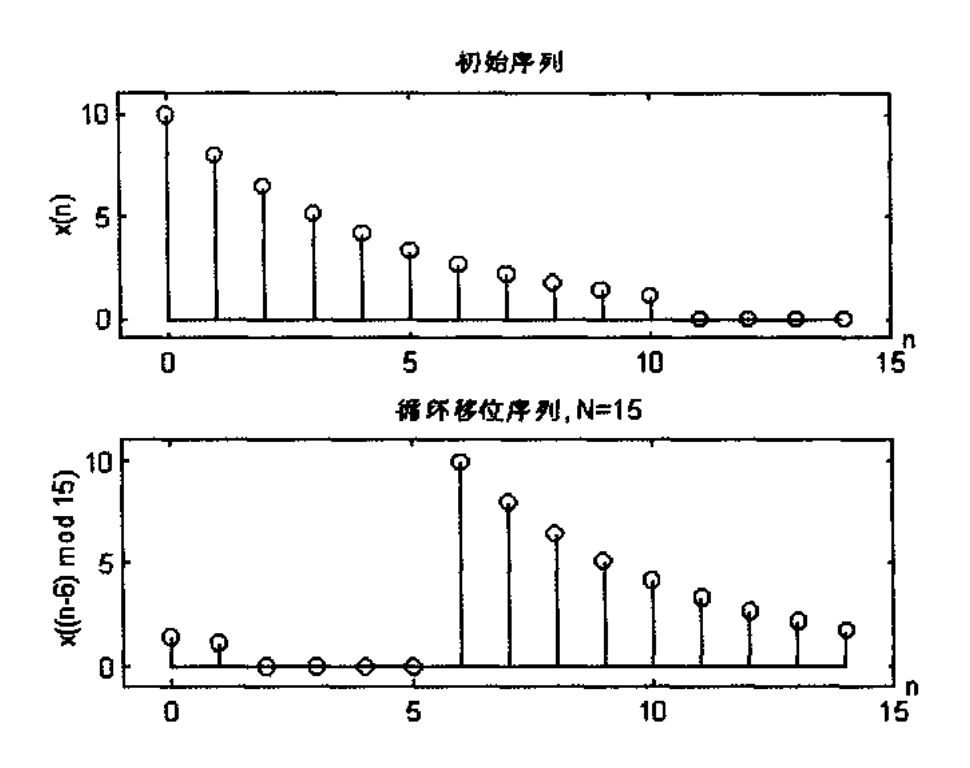


图 5 .17　例 5.12 的循环移位序列

6.频域中的循环移位:此性质与上一性质成对偶关系:

$$\mathrm{DFT}[W_N^{-ln}x(n)] = X((k-l))_N R_N(k) \tag{5.38}$$

7.循环卷积:两个 N 点序列的线性卷积会导致一更长的序列。我们再次将区间限制在 $0 \leqslant n \leqslant N-1$,现在考虑循环移位而不是线性移位,包含循环移位的卷积运算叫做循环卷积,定义为:

$$x_1(n) Ⓝ x_2(n) = \sum_{m=0}^{N-1} x_1(n) x_2((n-m))_N, \ 0 \leq n \leq N-1 \tag{5.39}$$

注意循环卷积结果仍为 N 点序列,它与线性卷积的结构类似。不同点在于求和范围及 N 点循环移位。它与 N 有关,也叫做 N 点循环卷积,因此采用符号Ⓝ表示是恰当的。循环卷积的 DFT 性质为:

$$\mathrm{DFT}[x_1(n) Ⓝ x_2(n)] = X_1(k) \cdot X_2(k) \tag{5.40}$$

此性质的另一种解释为:当在频域中进行两个 N 点 DFT 相乘时,在时域中得到循环卷积(而不是通常的线性卷积)。

例 5.13　设 $x_1(n) = \{1,2,2\}$, $x_2(n) = \{1,2,3,4\}$ 计算 4 点循环卷积 $x_1(n)④x_2(n)$。

解:注意 $x_1(n)$ 为 3 点序列,进行先进卷积之前必须在其尾部填一个零,使其成为 4 点序

列。我们将分别在时域及频域中求解这个问题,在时域中采用循环卷积技术,而在频域中采用DFT。

• 时域方法:4 点循环卷积由下式给出:

$$x_1(n)④x_2(n)=\sum_{m=0}^{3}x_1(m)x_2((n-m))_4$$

对每一个 n,我们必须产生一循环移位序列,将它的样本逐个与 $x_1(m)$ 相乘,然后求和,得此 n 值的循环卷积值,在 $0\leqslant n\leqslant 3$ 上重复此过程。考虑

$$x_1(m)=\{1,2,2,0\} \quad 和 \quad x_2(m)=\{1,2,3,4\}$$

$n=0$ 时

$$\begin{aligned}\sum_{m=0}^{3}x_1(m)\cdot x_2((0-m))_4 &= \sum_{m=0}^{3}[\{1,2,2,0\}\cdot\{1,4,3,2\}]\\ &= \sum_{m=0}^{3}\{1,8,6,0\}=15\end{aligned}$$

$n=1$ 时

$$\begin{aligned}\sum_{m=0}^{3}x_1(m)\cdot x_2((1-m))_5 &= \sum_{m=0}^{3}[\{1,2,2,0\}\cdot\{2,1,4,3\}]\\ &= \sum_{m=0}^{3}\{2,2,8,0\}=12\end{aligned}$$

$n=2$ 时

$$\begin{aligned}\sum_{m=0}^{3}x_1(m)\cdot x_2((2-m))_5 &= \sum_{m=0}^{3}[\{1,2,2,0\}\cdot\{3,2,1,4\}]\\ &= \sum_{m=0}^{3}\{3,4,2,0\}=9\end{aligned}$$

$n=3$ 时

$$\begin{aligned}\sum_{m=0}^{3}x_1(m)\cdot x_2((3-m))_5 &= \sum_{m=0}^{3}[\{1,2,2,0\}\cdot\{4,3,2,1\}]\\ &= \sum_{m=0}^{3}\{4,6,4,0\}=14\end{aligned}$$

因此

$$x_1(n)④x_2(n)=\{15,12,9,14\}$$

• 频域方法：首先计算 $x_1(n)$和 $x_2(n)$的 4 点 DFT，逐个样本相乘，取逆 DFT，得到循环卷积。

$x_1(n)$的 DFT

$$x_1(n)=\{1,2,2,0\}\Rightarrow X_1(k)=\{5,-1-j2,1,-1+j2\}$$

$x_2(n)$的 DFT

$$x_2(n)=\{1,2,3,4\}\Rightarrow X_2(k)=\{10,-2+j2,-2,-2-j2\}$$

然后

$$X_1(k)\cdot X_2(k)=\{50,6+j2,-2,6-j2\}$$

IDFT 后

$$x_1(n)④x_2(n)=\{15,12,9,14\}$$

与前结果相同。

与循环移位类似，我们可用不同的方法实现循环卷积。最简单的方法是用 cirshftt 函数和两个嵌套的 for...end 环实现(5.39)。显然，这是一种效率不高的算法。另一种算法是对$[0,N-1]$内的 n，产生序列 $x((n-m))_N$ 作为矩阵的行，类似于 dft 函数，我们用矩阵－矢量乘法实现(5.39)。这需要一个 for...end 环，下面的 circonvt 函数综合实现了这些步骤。

```
function y = circonvt(x1,x2,N)
% 在 x1 和 x2: (时域)之间的 N 点循环卷积
% ——————————————————————
% [y] = circonvt(x1,x2,N)
% y = 包含循环卷积的输出序列
% x1 = 长度 N1 <= N 的输入序列
% x2 = 长度 N2 <= N 的输入序列
% N = 循环缓冲器的大小
% 方法 y(n) = sum (x1(m) * x2((n-m) mod N))

% 检查 x1 的长度
if length(x1) > N
      error('N 必须 >= x1 的长度')
end
```

```
% 检查 x2 的长度
if length(x2) > N
        error('N 必须 > = x2 的长度')
end
x1 = [x1 zeros(1,N - length(x1))];
x2 = [x2 zeros(1,N - length(x2))];
m = [0:1:N - 1];
x2 = x2(mod( - m,N) + 1);
H = zeros(N,N);
for n = 1:1:N
H(n,:) = cirshftt(x2,n - 1,N);
end
y = x1 * H';
```

第三种方法则可利用 dft 函数实现频域运算(5.40)。这会在习题 5.15 中探讨。

例 5.14 用 MATLAB 实现例 5.13 中的循环卷积。

解:序列为:$x_1(n)=\{1,2,2\}$,$x_2(n)=\{1,2,3,4\}$

```
>> x1 = [1,2,2]; x2 = [1,2,3,4];
>> y = circonvt(x1,x2,4)
y =
    15    12    9    14
```

因此

$$x_1(n)④x_2(n)=\{15,12,9,14\}$$

与前相同。

例 5.15 在这个例子中,我们将研究 N 的大小对循环卷积的影响,显然 $N\geqslant4$;否则 $x_2(n)$将在时域中出现混叠,我们将采用与例 5.13 相同的两个序列。

a. 计算 $x_1(n)⑤x_2(n)$.

b. 计算 $x_1(n)⑥x_2(n)$.

c. 对所得结果进行讨论。

解:尽管序列为 $x_1(n)=\{1,2,2\}$,$x_2(n)=\{1,2,3,4\}$,与例 5.14 相同。但对于不同的 N,我们仍希望得到不同的结果。这与线性卷积的情形不同,那时给定两个序列,对于不同的 N,结果是唯一的。

a. 5 点循环卷积:

```
>> x1 = [1,2,2]; x2 = [1,2,3,4];
>> y = circonvt(x1,x2,5)
```

```
y =
     9     4     9     14     14
```

因此

$$x_1(n)⑤x_2(n)=\{9,4,9,14,14\}$$

b. 6点循环卷积：

```
>> x1 = [1,2,2]; x2 = [1,2,3,4];
>> y = circonvt(x1,x2,6)
y =
     1     4     9     14     14     8
```

因此

$$x_1(n)⑥x_2(n)=\{1,4,9,14,14,8\}$$

c.仔细观察此例和上一例中的4点、5点、6点循环卷积，会发现一些独特的特征。显然，N点循环卷积结果是一N点序列，这些卷积中的一些样本具有相同的值，而有些样本的值为其他卷积里样本的和。例如5点卷积中的第一个样本为6点卷积中的第一个样本和最后一个样本的和。而 $x_1(n)$，$x_2(n)$的线性卷积由下式给出：

$$x_1(n)*x_2(n)=\{1,4,9,14,14,8\}$$

与6点循环卷积的结果相同，这些与其他结果将在下节中讨论。

8.乘法性质 这与循环卷积性质成对偶关系，此性质由下式给出：

$$\mathrm{DFT}[x_1(n)\cdot x_2(n)]=\frac{1}{N}X_1(k)Ⓝ X_2(k) \tag{5.41}$$

它在频域中执行循环卷积。因为 $X_1(k)$，$X_2(k)$都是 N 点序列，用来计算循环卷积的MATLAB函数也可在这儿使用。

9.Parseval定理：用此定理可计算频域的能量：

$$E_x=\sum_{n=0}^{N-1}|x(n)|^2=\frac{1}{N}\sum_{k=0}^{N-1}|X(k)|^2 \tag{5.42}$$

$\frac{|X(k)|^2}{N}$叫做有限持续时间序列的能量频谱。类似地，对于周期信号，$\left|\frac{\tilde{X}(k)}{N}\right|^2$叫做功率频谱。

用 DFT 计算线性卷积

线性系统中最重要的一个运算是线性卷积。实际中的 FIR 滤波器通常是用线性卷积实现的。另一方面,DFT 是在频域中实现线性系统的实用方法。正如我们将要看到的,它在计算形式上,也是很有效的。但也存在一个问题。DFT 的结果是循环卷积而不是线性卷积。(在某种程度上,这不是我们所希望的。)怎样用 DFT 执行线性卷积(或者说,怎样使线性卷积与循环卷积等价)呢? 在例 5.15 中曾暗示过这个问题。

设 $x_1(n)$为 N_1 点序列,$x_2(n)$为 N_2 点序列,$x_3(n)$定义为 $x_1(n)$和 $x_2(n)$的线性卷积,即:

$$x_3(n) = x_1(n) * x_2(n)$$

$$= \sum_{k=-\infty}^{\infty} x_1(k)x_2(n-k) = \sum_{0}^{N_1-1} x_1(k)x_2(n-k) \tag{5.43}$$

因此 $x_3(n)$为(N_1+N_2-1)点序列。如果选择 $N=\max(N_1,N_2)$,计算 N 点循环卷积 $x_1(n)\textcircled{N}x_2(n)$,得到一个 N 点序列,它显然与 $x_3(n)$不同。这给我们一个启示,为什么不选择 $N=N_1+N_2-1$,进行 N_1+N_2-1 循环卷积呢? 至少这使两个卷积具有相同的样本数。

因此设 $N=N_1+N_2-1$,把 $x_1(n)$和 $x_2(n)$看成一个 N 点序列,定义 $x_4(n)$为 N 点循环卷积。

$$x_4(n) = x_1(n)\textcircled{N}x_2(n)$$

$$= \left[\sum_{m=0}^{N-1} x_1(k)x_2((n-m))_N\right]R_N(n)$$

$$= \left[\sum_{m=0}^{N-1} x_1(k)\sum_{r=-\infty}^{\infty} x_2(n-m-rN)\right]R_N(n)$$

$$= \left[\sum_{r=-\infty}^{\infty} \underbrace{\sum_{m=0}^{N_1-1} x_1(k)x_2(n-m-rN)}_{x_3(n-rN)}\right]R_N(n)$$

$$= \left[\sum_{r=-\infty}^{\infty} x_3(n-rN)\right]R_N(n) \tag{5.44}$$

这一分析说明,一般地讲,循环卷积是线性卷积的一种混叠形式。例 5.15 已说明了这一点。

既然 $x_3(n)$是 $N=(N_1+N_2-1)$点序列,则有

$$x_4(n) = x_3(n);\ 0 \le n \le (N-1)$$

这就意味着在时域中不存在混叠。

结论：若通过填充适当的零数使 $x_1(n)$ 和 $x_2(n)$ 成为 $N = N_1 + N_2 - 1$ 序列，那么循环卷积与线性卷积相等。

例 5.16 设 $x_1(n)$ 和 $x_2(n)$ 是两个 4 点序列：

$$x_1(n) = \{1,2,2,1\},\ x_2(n) = \{1,-1,1,-1\}$$

a. 确定它们的线性卷积 $x_3(n)$。

b. 计算循环卷积 $x_4(n)$，证明它与 $x_3(n)$ 相等。

解： 我们用 MATLAB 来解决这个问题。

a. MATLAB 程序

```
>> x1 = [1,2,2,1];x2 = [1,-1,1,-1];
>> x3 = conv(x1,x2)
x3 =    1    1    -1    -2    -1    1    1
```

于是线性卷积 $x_3(n)$ 为 7 点序列：

$$x_3(n) = \{1,1,-1,-2,-1,1,1\}$$

b. 必须使 $N \geqslant 7$，选择 $N = 7$，则有：

```
>> x4 = circonvt(x1,x2,7)
x4 =  1    1    -1    -2    -1    1    1
```

因此

$$x_4(n) = \{1,1,-1,-2,-1,1,1\} = x_3(n)$$

误差分析

为了用 DFT 作线性卷积，必须选择适当的 N，但在实际中却可能做不到这一点，尤其当 N 很大而存储空间却有限时。当选择比所需要的 N 小的值进行循环卷积时，就引入了误差。在实际中它是很有用的，因此需要计算此误差。显然，$N \geqslant \max(N_1, N_2)$，设

$$\max(N_1, N_2) \leqslant N < (N_1 + N_2 - 1)$$

由前面的分析(5.44)可得：

$$x_4(n)=\left[\sum_{r=-\infty}^{\infty} x_3(n-rN)\right]R_N(n)$$

则误差为：

$$\begin{aligned} e(n) &\triangleq x_4(n)-x_3(n) \\ &=\left[\sum_{r\neq 0} x_3(n-rN)\right]R_N(n) \end{aligned}$$

因为 $N \geq \max(N_1, N_2)$，上面的求和式中只剩下对应于 $r=\pm 1$ 的两项。因此

$$e(n)=[x_3(n-N)+x_3(n+N)]R_N(n)$$

一般来讲，$x_1(n)$和 $x_2(n)$为因果序列，因此 $x_3(n)$也为因果序列，即：

$$x_3(n-N)=0;\ 0\leqslant n\leqslant N-1$$

因此

$$e(n)=x_3(n+N);\ 0\leqslant n\leqslant N-1$$

这是一个简单但很重要的关系。它说明当 $\max(N_1, N_2)\leqslant N<(N_1+N_2-1)$时，样本 n 处的误差与线性卷积中第 $n+N$ 个样本相等。(N_1+N_2-1)个样本后，线性卷积结果为零。这说明循环卷积中的头几个样本存在误差，而剩下的则为正确的线性卷积样本。

例 5.17 考察上例中的序列 $x_1(n)$和 $x_2(n)$，计算当 $N=6,5,4$ 时的循环卷积。在每种情形下验证误差关系。

解：显然，线性卷积 $x_3(n)$仍相同。

$$x_3(n)=\{1,1,-1,-2,-1,1,1\}$$

当 $N=6$ 时，得到 6 点序列：

$$x_4(n)=x_1(n)⑥x_2(n)=\{2,1,-1,-2,-1,1\}$$

因此

$$\begin{aligned} e(n) &=\{2,1,-1,-2,-1,1\}-\{1,1,-1,-2,-1,1\}\ 0\leqslant n\leqslant 5 \\ &=\{1,0,0,0,0,0\} \\ &=x_3(n+6) \end{aligned}$$

与预想的相同。当 $N=5$ 时，得到5点序列：

$$x_4(n)=x_1(n)⑤x_2(n)=\{2,2,-1,-2,-1\}$$

$$\begin{aligned}e(n)&=\{2,2,-1,-2,-1\}-\{1,1,-1,-2,-1\}\ 0\leqslant n\leqslant 4\\&=\{1,1,0,0,0\}\\&=x_3(n+5)\end{aligned}$$

最后，当 $N=4$ 时，得到4点序列：

$$x_4(n)=x_1(n)④x_2(n)=\{0,2,0,-2\}$$

$$\begin{aligned}e(n)&=\{0,2,0,-2\}-\{1,1,-1,-2\}\ 0\leqslant n\leqslant 3\\&=\{-1,1,1,0\}\\&=x_3(n+4)\end{aligned}$$

$N=4$ 的最后一种情形也提供了一个有用的结论。

结论：当选择 $N=\max(N_1,N_2)$ 作循环卷积时，前 $(M-1)$ 个样本存在误差（即不同于线性卷积），其中 $M=\min(N_1,N_2)$。此结果在用分段处理法实现长卷积时是非常有用的。

分段卷积

当我们想对一个连续接收的输入序列（像从麦克风得到的语音信号）进行滤波时，出于实用目的，应将此信号看作无限持续时间序列。若想用 FIR 滤波器来实现滤波，它的 DFT 是用线性卷积计算的，则会遇到一些实际问题。我们必须计算一个巨大的 DFT，一般来讲，这是不切实际的。而且，需要处理所有的输入样本后，输出样本才有效，这会引进不能忍受的大延迟。因此必须把无限长度的序列分成较小的几部分（或段），用 DFT 处理每一部分，然后把每段的输出装配起来，成为输出序列。此过程叫做分段卷积（或分段处理）。

假设把序列 $X(n)$ 分成多段 N 点序列，滤波器的脉冲响应为 M 点序列，$M<N$。由上面的结论我们注意到输入段和脉冲响应之间的循环卷积产生该段的输出序列，其中前 $(M-1)$ 个样本不是正确的输出值。若将 $X(n)$ 简单地分成互不重叠的各段，则所得的输出序列会有不正确样本的区间存在。为了纠正这个问题，把 $X(n)$ 分成每段与前一段重迭 $(M-1)$ 个样本的多段，保留后面的 $(N-M+1)$ 个输出样本，最后把这些输出连接成一个序列。为了修正第一段中的前 $(M-1)$ 个样本，把第一输入段的前 $(M-1)$ 个样本置为零。此过程叫做分段卷积的重叠保留法。显然，当 $N>>M$ 时，此方法更有效。我们将用简单的例子证明这一点。

例5.18 设 $x(n)=(n+1),0\leqslant n\leqslant 9$，$h(n)=\{\underset{\uparrow}{1},0,-1\}$，按 $N=6$ 用重叠保留法计算 $y(n)=x(n)*h(n)$。

解：由于 $M=3$，必须使每一段与前一段相重两个样本，$X(n)$ 为10点序列，我们需要在开头加 $(M-1)=2$ 个零。因为 $N=6$，则可划为三部分：

$$x_1(n)=\{0,0,1,2,3,4\}$$
$$x_2(n)=\{3,4,5,6,7,8\}$$
$$x_3(n)=\{7,8,9,10,0,0\}$$

因为 $x(n)$在 $n=9$ 外无值,因此在 $x_3(n)$中必须填两个零。现在计算每一部分与 $h(n)$的循环卷积。

$$y_1=x_1(n)⑥h(n)=\{-3,-4,1,2,2,2\}$$
$$y_2=x_2(n)⑥h(n)=\{-4,-4,2,2,2,2\}$$
$$y_3=x_3(n)⑥h(n)=\{7,8,2,2,-9,-10\}$$

注意丢弃前两个样本,装配输出 $y(n)$为:

$$y(n)=\{\underset{\uparrow}{1},2,2,2,2,2,2,2,2,2,-9,-10\}$$

线性卷积为:

$$x(n)*h(n)=\{\underset{\uparrow}{1},2,2,2,2,2,2,2,2,2,-9,-10\}$$

与重叠保留法结果相符。

MATLAB 实现

以上述例子为指导,我们可以探讨一个 MATLAB 函数实现长序列 $x(n)$的重叠保留法。此函数中关键的一步是为找到一个为分段定出下标的恰当的方法。当 $n\geqslant0$ 时给定 $x(n)$,在分段处理前,先将头$(M-1)$个样本置零。扩大后的序列为:

$$\hat{x}(n)\triangleq\{\underbrace{0,0,\ldots,0}_{(M-1)\text{个零}},x(n)\},\ n\geq0$$

设 $L=N-M+1$,则第 k 段 $x_k(n)$, $0\leqslant n\leqslant N-1$,为:

$$x_k(n)=\hat{x}(m);\ kL\leq m\leq kL+N-1,k\geq0,0\leq n\leq N-1$$

所分的段数为:

$$K=\left[\frac{N_x+M-2}{L}\right]+1$$

其中 N_x 是 $x(n)$的长度,$\lfloor\cdot\rfloor$是截取运算。现在可用早先探讨的 circonvt 函数计算每一段与 $h(n)$的循环卷积:

$$y_k(n) = x_k(n) \textcircled{N} h(n)$$

最后,丢弃每一个 $y_k(n)$中的前$(M-1)$个样本,连接剩下的样本,得到线性卷积 $y(n)$。这过程由下面的 ovrlpsav 函数具体实现。

```
function [y] = ovrlpsav(x,h,N)
% 用重叠保留法作分段卷积
% ————————————————
% [y] = ovrlpsav(x,h,N)
% y = 输出序列
% x = 输入序列
% h = 脉冲响应
% N = 段长
%
Lenx = length(x); M = length(h);
M1 = M - 1; L = N - M1;
h = [h zeros(1,N - M)];
%
x = [zeros(1,M1), x, zeros(1,N - 1)];   % 预置 (M - 1) 个零
K = floor((Lenx + M1 - 1)/(L));         %段数
Y = zeros(K + 1,N);
% 与后续各段卷积
for k = 0:K
xk = x(k * L + 1:k * L + N);
Y(k + 1,:) = circonvt(xk,h,N);
end
Y = Y(:,M:N)';                          % 去掉前(M - 1) 个样本
y = (Y(:))';                            % 装成输出
```

应指出一点,这里探讨的 ovrlpsav 函数不是最有效的方法。当讨论快速付利叶变换时,我们将重新讨论这个问题。

例 5.19 为了验证 ovrlpsav 函数,考察例 5.18 中的序列。

解:MATLAB 程序————————————————————————

```
>> n = 0:9; x = n + 1; h = [1,0, - 1]; N = 6;
>> y = ovrlpsav(x,h,N)
y =
```

1	2	2	2	2	2	2	2	2	2	-9	-10

此即为正确的线性卷积,与预想结果一样。

另一种分段卷积的方法是重叠相加法。在此方法中,把输入序列分为互不重叠的段,与脉冲响应卷积,所得的输出段与后来的各段重叠相加形成最终的输出。这会在习题 5.20 中探讨。

快速付利叶变换

前面介绍的 DFT(5.24)是时域和频域均为离散的唯一的一种变换,它适用于有限持续时间序列。尽管它是数值可计算的,若直接实现(5.24),则效率极低,尤其当序列长度很大时。1965 年,Cooley 与 Tukey[4]提出了大幅度减少 DFT 中的计算量的方法,这导致了包括数字信号处理领域在内的 DFT 应用的兴起,同时,它也引起其他有效算法的探讨。所有这些有效算法统称为快速付利叶变换。

考虑 N 点序列 $x(n)$,它的 N 点 DFT 可由(5.24)计算,这里重写如下:

$$X(k)=\sum_{n=0}^{N-1} x(n)W_N^{nk}, \quad 0\leqslant k\leqslant N-1 \tag{5.46}$$

其中 $W_N=e^{-j2\pi/N}$。为了得到 $X(k)$的一个样本,需要进行 N 次复数乘法和$(N-1)$次复数加法。因此为得到全部的 DFT 系数,需要进行 N^2 次复数乘法和 $N(N-1)$次复数加法。同时也要存储 N^2 个复数系数 $\{W_N^{nk}\}$(或者须付出额外开销由内部产生)。显然,N 点序列的 DFT 计算量依赖于 N^2,由下式来表示:

$$C_N=o(N^2)$$

对于大 N,$o(N^2)$在实际中是不能接受的。一般地,一次加法比一次乘法的所需时间要短得多。因此,从现在起,我们重点考虑复数乘法的次数,一次复数乘法需要 4 次实数乘法和两次加法。

有效算法的目标 在有效设计的算法中,每个数据样本的计算量应为常数,因此总的计算量应该与 N 成线性关系。

利用$\{W_N^{nk}\}$的周期性

$$W_N^{kn}=W_N^{k(n+N)}=W_N^{(k+N)n}$$

和对称性

$$W_N^{kn+N/2}=-W_N^{kn}$$

可削减依赖于 N 平方的大部分计算量。

只考虑 W_N^{nk} 周期性的算法叫做戈泽尔(Goertzel)算法。这种算法仍旧需要 $C_N=o(N^2)$ 次乘法,但它具有一定的优点。在第十章将讲述这种算法。我们首先通过例子说明利用对称性和周期性减少计算量的优越性,接着分析两种具体的 FFT 算法,它们需要 $C_N=o(N\log N)$ 次计算。这就是时间抽选算法(DIT-FFT)和频率抽选算法(DIF-FFT)。

例 5.20　讨论 4 点 DFT,并探讨出一种有效算法简化计算:

$$X(k)=\sum_{n=0}^{3}x(n)W_4^{nk},\ 0\leqslant n\leqslant 3;\ W_4=e^{-j2\pi/4}=-j$$

解:上述计算写成矩阵形式为:

$$\begin{bmatrix}X(0)\\X(1)\\X(2)\\X(3)\end{bmatrix}=\begin{bmatrix}W_4^0 & W_4^0 & W_4^0 & W_4^0\\ W_4^0 & W_4^1 & W_4^2 & W_4^3\\ W_4^0 & W_4^2 & W_4^4 & W_4^6\\ W_4^0 & W_4^3 & W_4^6 & W_4^9\end{bmatrix}\begin{bmatrix}x(0)\\x(1)\\x(2)\\x(3)\end{bmatrix}$$

需要 16 次复数乘法。

高效算法:利用周期性:

$$W_4^0=W_4^4=1\quad;\quad W_4^1=W_4^9=-j$$
$$W_4^2=W_4^6=-1\quad;\quad W_4^3=j$$

代入上述的矩阵,得到:

$$\begin{bmatrix}X(0)\\X(1)\\X(2)\\X(3)\end{bmatrix}=\begin{bmatrix}1 & 1 & 1 & 1\\ 1 & -j & -1 & j\\ 1 & -1 & 1 & -1\\ 1 & j & -1 & -j\end{bmatrix}\begin{bmatrix}x(0)\\x(1)\\x(2)\\x(3)\end{bmatrix}$$

利用对称性,有:

$$X(0)=x(0)+x(1)+x(2)+x(3)=[\underbrace{x(0)+x(2)}_{g_1}]+[\underbrace{x(1)+x(3)}_{g_2}]$$

$$X(1)=x(0)-jx(1)-x(2)+jx(3)=[\underbrace{x(0)-x(2)}_{h_1}]-j[\underbrace{x(1)-x(3)}_{h_2}]$$

$$X(2)=x(0)-x(1)+x(2)-x(3)=[\underbrace{x(0)+x(2)}_{g_1}]-[\underbrace{x(1)+x(3)}_{g_2}]$$

$$X(3)=x(0)+jx(1)-x(2)-jx(3)=[\underbrace{x(0)-x(2)}_{h_1}]+j[\underbrace{x(1)-x(3)}_{h_2}]$$

因此高效算法为：

步骤 1	步骤 2
$g_1=x(0)+x(2)$	$X(0)=g_1+g_2$
$g_2=x(1)+x(3)$	$X(1)=h_1-jh_2$
$h_1=x(0)-x(2)$	$X(2)=g_1-g_2$
$h_2=x(1)-x(3)$	$X(3)=h_1+jh_2$

(5.47)

只需要两次复数乘法，甚至对这个简单例子来讲，它都减少到了一个相当小的数。此算法的信号流图如图 5.18 所示。

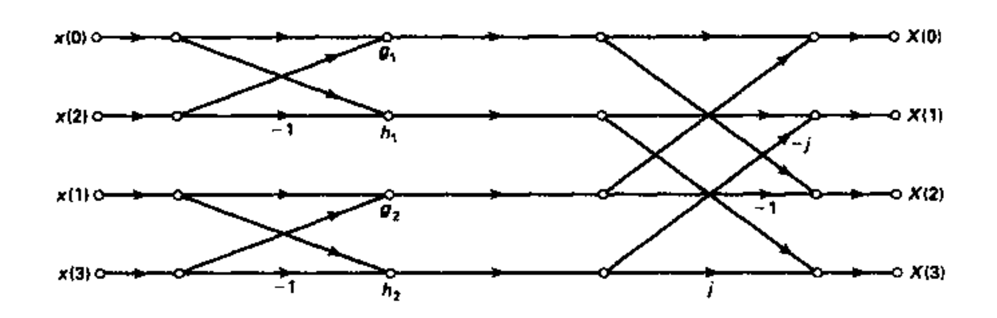


图 5.18 例 5.20 的信号流图

解释：高效算法(5.47)可有多种解释。首先，4 点序列 $x(n)$ 可分成两个 2 点序列，排成列向量形式为：

$$\left[\begin{bmatrix}x(0)\\x(2)\end{bmatrix},\begin{bmatrix}x(1)\\x(3)\end{bmatrix}\right]=\begin{bmatrix}x(0)&x(1)\\x(2)&x(3)\end{bmatrix}$$

接着，对每列进行 2 点 DFT：

$$W_2\begin{bmatrix}x(0)&x(1)\\x(2)&x(3)\end{bmatrix}=\begin{bmatrix}1&1\\1&-1\end{bmatrix}\begin{bmatrix}x(0)&x(1)\\x(2)&x(3)\end{bmatrix}$$

$$=\begin{bmatrix} x(0)+x(2) & x(1)+x(3) \\ x(0)-x(2) & x(1)-x(3) \end{bmatrix}=\begin{bmatrix} g_1 & g_2 \\ h_1 & h_2 \end{bmatrix}$$

然后所得矩阵的每个元素被$\{W_4^{pq}\}$相乘,其中 p 为列标,q 为行标;也就是进行了点积运算:

$$\begin{bmatrix} 1 & 1 \\ 1 & -j \end{bmatrix} * \begin{bmatrix} g_1 g_2 \\ h_1 h_2 \end{bmatrix}=\begin{bmatrix} g_1 & g_2 \\ h_1 & -jh_2 \end{bmatrix}$$

最后,计算两个行向量的 2 点 DFT:

$$\begin{bmatrix} g_1 & g_2 \\ h_1 & -jh_2 \end{bmatrix} W_2=\begin{bmatrix} g_1 & g_2 \\ h_1 & -jh_2 \end{bmatrix}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}=\begin{bmatrix} g_1+g_2 & g_1-g_2 \\ h_1-jh_2 & h_1+jh_2 \end{bmatrix}$$
$$=\begin{bmatrix} X(0) & X(2) \\ X(1) & X(3) \end{bmatrix}$$

尽管这种解释使用的乘法似乎比高效算法多,但它确实提供了基于较小 DFT 计算大 DFT 的方法。

划分-组合法

为了减小 DFT 的与 N 的平方成比例的计算量,必须选择合数 $N = LM$,因为:

$$L^2 + M^2 \ll N^2 \quad \text{当 } N \text{ 较大时}$$

现在,把序列分成长度为 L 的 M 个序列,进行 M 次 L 点 DFT,然后用 L 个较小的 M 点 DFT 组合成大的 DFT,这是划分-组合法的基本思想。设 $N = LM$,(5.46)中的 n 与 k 可写为:

$$\begin{aligned} n &= Ml + m, \quad 0 \leqslant l \leqslant L-1,\ 0 \leqslant m \leqslant M-1 \\ k &= p + Lq, \quad 0 \leqslant p \leqslant L-1,\ 0 \leqslant q \leqslant M-1 \end{aligned} \tag{5.48}$$

把序列 $x(n)$和 $X(k)$分别改写为数组 $x(l,m)$和 $X(p,q)$,则(5.46)可写成:

$$\begin{aligned} X(p,q) &= \sum_{m=0}^{M-1}\sum_{l=0}^{L-1} x(l,m)\, W_N^{(Ml+m)(p+Lq)} \\ &= \sum_{m=0}^{M-1}\left\{ W_N^{mp}\left[\sum_{l=0}^{L-1} x(l,m)\, W_N^{Mlp}\right]\right\} W_N^{Lmq} \end{aligned} \tag{5.49}$$

$$= \underbrace{\sum_{m=0}^{M-1}\left\{W_N^{mp}\left[\underbrace{\sum_{l=0}^{L-1} x(l,m) W_L^{lp}}_{L-点DFT}\right]\right\} W_M^{mq}}_{M-点DFT}$$

因此可分三步实现(5.49)

1.第一步,对每一列 $m=0,\ldots,M-1$,计算 L 点 DFT。

$$F(p,m)=\sum_{l=0}^{L-1} x(l,m) W_L^{lp};\ 0\leqslant p\leqslant L-1 \tag{5.50}$$

2.第二步,修正 $F(p,m)$,得到另一数组。

$$G(p,m)=W_N^{pm}F(p,m),\quad \begin{matrix}0\leqslant p\leqslant L-1\\0\leqslant m\leqslant M-1\end{matrix} \tag{5.51}$$

其中 W_N^{pm} 叫做转动因子。

3.最后,对每一行 $p=0,\ldots,L-1$,计算 M 点 DFT。

$$X(p,q)=\sum_{m=0}^{M-1} G(p,m) W_M^{mq}\quad 0\leqslant q\leqslant M-1 \tag{5.52}$$

此方法所用的复数乘法次数为:

$$C_N=ML^2+N+LM^2<o(N^2) \tag{5.53}$$

若 M 或 L 为合数,可重复此过程。显然,当 N 是一个高度合数时,可得到最有效的算法。$N=R^{\nu}$ 时,这样的算法叫做基-R FFT 算法。$N=R_1^{\nu 1}R_2^{\nu 2}\cdots$时,叫做混合基算法。最受欢迎的,也是易编程的算法是基-2 FFT 算法。

基-2 FFT 算法

若 $N=2^{\nu}$,则选择 $M=2$, $L=N/2$,依照(5.48)把 $x(n)$分成两个 $N/2$ 点序列:

$$\begin{matrix}g_1(n)=x(2n)\\g_2(n)=x(2n+1)\end{matrix};\ 0\leqslant n\leqslant\frac{N}{2}-1$$

序列 $g_1(n)$包含 $x(n)$中的偶序样本,$g_2(n)$包含 $x(n)$中的奇序样本。设 $G_1(k)$、$G_2(k)$为 $g_1(n)$和 $g_2(n)$的 $N/2$ 点 DFT。由(5.49)得:

$$X(k) = G_1(k) + W_N^k G_2(k),\ 0 \leq k \leq N-1 \tag{5.54}$$

这叫做融合公式,它把两个 $N/2$ 点 DFT 组合成一个 N 点 DFT。所用复数乘法次数减少为:

$$C_N = \frac{N^2}{2} + N = o(N^2/2)$$

此过程可一直重复下去,每一次划分序列,把较小的 DFT 组合起来。经过 ν 次后,得到 N 个 1 点序列,也为 1 点 DFT。上述过程叫做时间抽选 FFT(DIT-FFT)算法,它所用复数乘法次数为:

$$C_N = N\nu = N\log_2 N$$

显然,若 N 很大时,C_N 与 N 近似成线性关系,这是采用高效算法要达到的目标。若再利用对称性,C_N 能减至$\frac{N}{2}\log_2 N$。$N=8$ 时,此算法的信号流图如图 5 .19 所示。

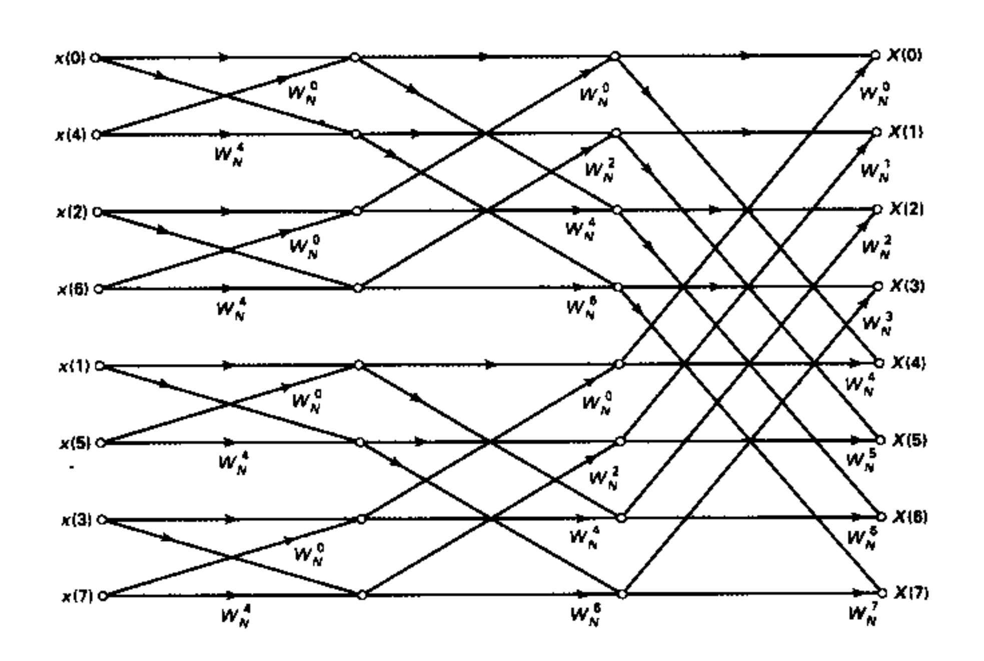


图 5 .19 $N=8$ 时的时间抽选 FFT 结构

另一种方法为:选择 $L=2, M=N/2$,然后依照(5.49)进行计算。注意初始 DFT 为 2 点 DFT,它不包含复数乘法。由(5.50)得:

$$F(0,m) = x(0,m) + x(1,m)W_2^0$$
$$= x(n) + x(n+N/2),0 \leq n \leq N/2$$
$$F(1,m) = x(0,m) + x(1,m)W_2^1$$

$$= x(n) - x(n+N/2), 0 \leqslant n \leqslant N/2$$

由(5 .51)得：

$$\begin{aligned} G(0,m) &= F(0,m) W_N^0 \\ &= x(n) + x(n+N/2), 0 \leqslant n \leqslant N/2 \\ G(1,m) &= F(1,m) W_N^m \\ &= [x(n) - x(n+N/2)] W_N^n, 0 \leqslant n \leqslant N/2 \end{aligned} \tag{5.55}$$

设 $G(0,m) = d_1(n), G(1,m) = d_2(n), 0 \leqslant n \leqslant N/2 - 1$ (既然它们可看作时域序列)；由(5.52)可得：

$$\begin{aligned} X(0,q) &= X(2q) = D_1(q) \\ X(1,q) &= X(2q+1) = D_2(q) \end{aligned} \tag{5.56}$$

这意味着以抽选的方式计算 DFT 的值 $X(k)$，因此这种算法叫做频率抽选(DIF-FFT)算法。它的信号流图是 DIT-FFT 结构的转置，它的计算量也等于$\frac{N}{2}\log_2 N$。

MATLAB 实现

MATLAB 提供 fft 函数来计算矢量 x 的 DFT。计算 N 点 DFT 的调用形式为 X = fft(x, N)。如果 x 的长度小于 N，则在其后填零使其成为 N 点序列。如果省略变量 N，则 DFT 的长度即为 x 的长度。如果 x 为一个矩阵，则计算 x 中每列的 N 点 DFT。

fft 函数是用机器语言，而不是以 MATLAB 指令写成的(即不存在 .m 文件)，因此它的执行速度很快，它采用混合基算法。如果 N 为 2 的幂，则得到高速的基 - 2 FFT 算法；若 N 不是 2 的乘方，则将 N 分解成质数，得到较慢的混合基 FFT 算法；最后，若 N 为质数，则 fft 函数采用的是原始 DFT 算法。

逆 DFT 由 ifft 函数计算，它的特征与 fft 函数相同。

例 5.21 此例研究当 $1 \leqslant n \leqslant 2048$ 时，fft 函数的执行时间，这将展现不同的 N 的情形下，划分-组合的战略。

解：MATLAB 提供了两个函数来确定执行时间。clock 函数读取瞬时时钟，etime(t1, t2) 函数计算时刻 t1, t2 间所经历的时间。为了确定执行时间，产生长度为 1 至 2048 的随机矢量，计算它们的 FFT，将计算时间存在一个数组里。最后，画出执行时间相对于 N 的图。

MATLAB 程序————————————————————

```
>> Nmax = 2048;
>> fft_time = zeros(1, Nmax);
>> for n = 1:1:Nmax
```

```
>>      x = rond(1,n);
>>      t = clock;fft(x);fft_time(n) = etime(clock,t);
>> end
>> n = [1:1:Nmax];
>> plot(n,fft_time,'.');
>> xlabel('N');ylabel('时间(单位:秒)')
>> title('FFT 执行时间')
```

执行时间曲线图如图 5.20 所示。这个图很说明问题,图中的点并不能用一个明确的函数表示,它们以不同的趋势分成几组,最上面一组描绘 $o(N^2)$的情形,这意味着这些 N 值肯定是 1 到 2048 间的质数,FFT 算法退化为 DFT 算法。类似地,也有相应于 $o(N^2/2)$,$o(N^2/3)$,$o(N^2/4)$,等等的组,说明 N 只经过很少的几次分解。最后一组表明(几乎线性)$o(N\log N)$的情形,其中 $N=2^\nu$,$0\leqslant\nu\leqslant 11$。对于这些 N,所用的是基-2 FFT 算法。对其他的 N,采用的是混合基 FFT 算法。这说明当 N 高度可合时,划分-组合策略是很有效的。例如 $N=2048$ 时,执行时间为 0.16 秒,$N=2047$ 时为 2.48 秒,$N=2039$ 时为 46.96 秒。

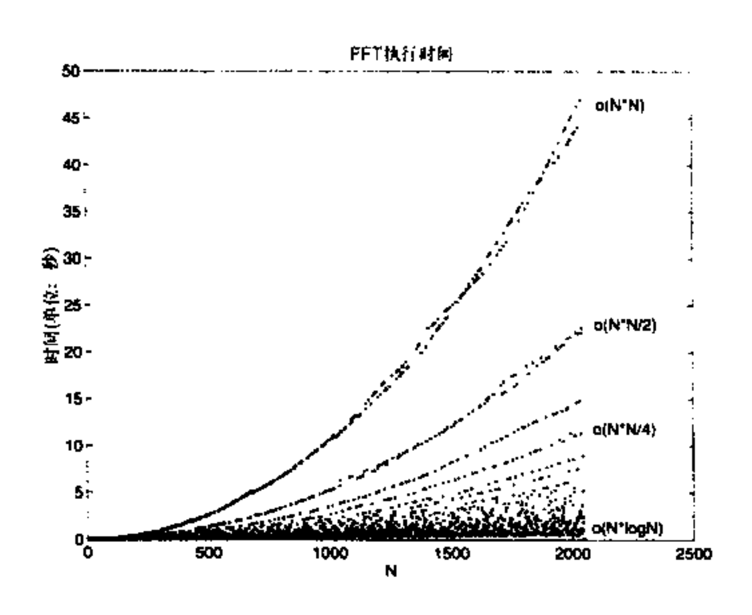


图 5.20 $1\leqslant n\leqslant 2048$ 的 FFT 执行时间

现在,可用 fft 函数代替 dft 函数,改进本章中早先探讨的 MATLAB 函数。从上例可知,必须慎用高度合数。除特定需求外,实际中的最佳选择是 $N=2^\nu$。

快速卷积

MATLAB 中用 filter(以 C 的形式写的)函数实现 conv 函数,对于较小的 N 值(<50),它是非常有效的。对于较大的 N 值,则可用 FFT 算法加速卷积。这种方法用循环卷积实现线性卷积,而用 FFT 实现循环卷积。所得的算法叫做快速卷积算法。另外,若选择 $N=2^\nu$,实现基-2FFT,则此算法叫做高速卷积算法。设 $x_1(n)$为 N_1 点序列,$x_2(n)$为 N_2 点序列,为加

速高速卷积,选择 N 为:

$$N = 2^{\lceil \log_2(N_1 + N_2 - 1) \rceil} \tag{5.57}$$

其中$\lceil x \rceil$为大于 x 的最小整数(也叫做上限函数)。现在线性卷积 $x_1(n) * x_2(n)$可通过计算两个 N 点 FFT,一个 N 点 IFFT 和一次 N 点点积得到。

$$x_1(n) * x_2(n) = \text{IFFT}[\text{FFT}[x_1(n)] \cdot \text{FFT}[x_2(n)]] \tag{5.58}$$

正如下例所示,对于大的 N,(5.58)比在时域中计算卷积快。

例 5.22 为了证明高速卷积的有效性,让我们比较两种方法的执行时间,设 $x_1(n)$为在区间[0,1]上均匀分布的 L 点随机序列,$x_2(n)$是均值为零,方差为 1 的 L 点高斯分布随机序列。通过计算 100 次随机序列的卷积,确定当 $1 \leqslant L \leqslant 150$ 时的卷积计算平均时间。

解:MATLAB 程序

```
conv_time = zeros(1,150); fft_time = zeros(1,150);
%
for L = 1:150
    tc = 0; tf = 0;
    N = 2 * N - 1; nu = ceil(log10(N1)/log10(2)); L = 2^nu;
    for I = 1:100
        h = randn(1,L);
        x = rand(1,L);
        t0 = clock; y1 = conv(h,x); t1 = etime(clock,t0);
        tc = tc + t1;
        t0 = clock; y2 = ifft(fft(h,N). * fft(x,N)); t2 = etime(clock,t0);
        tf = tf + t2;
    end
%
    conv_time(L) = tc/100;
    fft_time(L) = tf/100;
end
%
n = 1:150; subplot(1,1,1);
plot(n(25:150),conv_time(25:150),n(25:150),fft_time(25:150))
```

图 5.21 给出当 $25 \leqslant L \leqslant 150$ 时线性卷积与高速卷积的计算时间。必须注意,这些时间受执行 MATLAB 程序的计算平台影响。图 5.21 是在 33MHz 486 机上得到的。图中表明对于较小的 L,线性卷积要快一些,在 $N = 50$ 时出现交叉点,在此点之外,线性卷积时间按指数增

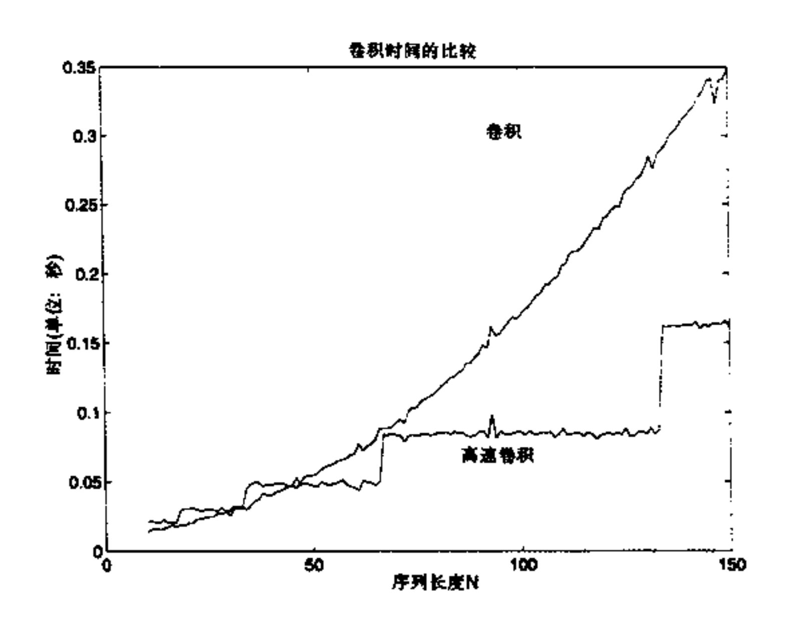


图 5.21 线性卷积与高速卷积的计算时间比较

长，而高速卷积时间则相当线性地增加。注意既然 $N=2\nu$，在 L 的一定范围内，高速卷积的执行时间为常数。

高速分段卷积

早先我们讨论过分段卷积算法，重叠保留法和重叠相加法，它们用相对较小的序列计算长序列的卷积。那一节中探讨的 ovrlpsav 函数采用 DFT 实现线性卷积，现在用基 - 2FFT 算法代替 DFT，得到高速重叠保留算法。为进一步缩减计算量，较短的(固定的)序列的 FFT 可只计算一次。下面的 hsolpsav 函数实现了这种算法。

```
function [y] = hsolpsav(x,h,N)
% 利用 FFT 进行的分段卷积的高速重叠保留法
% ----------------------------------------
% [y] = hsolpsav(x,h,N)
% y = 输出序列
% x = 输入序列
% h = 脉冲响应
% N = 段长(必须是二的幂次)
%
N = 2^(ceil(log10(N)/log10(2)));
Lenx = length(x); M = length(h);
M1 = M - 1; L = N - M1;
h = fft(h,N);
%
x = [zeros(1,M1), x, zeros(1,N-1)];
```

```
K = floor((Lenx + M1 - 1)/(L)); % 段的数目
Y = zeros(K + 1,N);
for k = 0:K
xk = fft(x(k * L + 1:k * L + N));
Y(k + 1,:) = real(ifft(xk. * h));
end
Y = Y(:,M:N)'; y = (Y(:))';
```

对重叠相加算法可作类似的改进。

习　　题

P5.1　用DFS定义求出下列周期序列的DFS系数,并用MATLAB验证。

a. $\tilde{x}_1(n)=\{2,0,2,0\},N=4$

b. $\tilde{x}_2(n)=\{0,0,1,0,0\},N=5$

c. $\tilde{x}_3(n)=\{3,-3,3,-3\},N=4$

d. $\tilde{x}_4(n)=\{j,j,-j,-j\},N=4$

e. $\tilde{x}_5(n)=\{1,j,j,1\},N=4$

P5.2　根据下列周期DFS系数,求出周期序列,首先用IDFS定义计算,然后用MATLAB验证。

a.　$\tilde{X}_1(k)=\{5,-2j,3,2j\},N=4$

b.　$\tilde{X}_2(k)=\{4,-5,3,-5\},N=4$

c.　$\tilde{X}_3(k)=\{1,2,3,4,5\},N=5$

d.　$\tilde{X}_4(k)=\{0,0,2,0\},N=4$

e.　$\tilde{X}_5(k)=\{0,j,-2j,-j\},N=4$

P5.3　设 $\tilde{x}_1(n)$的基本周期 $N=50$,它的一个周期由下式给出:

$$\tilde{x}_1(n)=\begin{cases} ne^{-0.3n}, & 0\leqslant n\leqslant 25 \\ 0, & 26\leqslant n\leqslant 49 \end{cases}$$

$\tilde{x}_2(n)$的基本周期 $N=100$,它的一个周期由下式给出

$$\tilde{x}_2(n)=\begin{cases} ne^{-0.3n}, & 0\leqslant n\leqslant 25 \\ 0, & 26\leqslant n\leqslant 99 \end{cases}$$

这两个周期序列的区别在于它们的周期,但它们具有相同的非零样本。

a.求出 $\tilde{x}_1(n)$的 DFS $\tilde{X}_1(k)$,用 stem 函数画出它的幅度和角度样本。

b.求出 $\tilde{x}_2(n)$的 DFS $\tilde{X}_2(k)$,画出它的幅度和角度样本。

c.上面两个 DFS 图有什么不同?

P5.4 考虑习题 5.3 中的 $\tilde{x}_1(n)$,设 $\tilde{x}_3(n)$是周期为 100 的周期序列,它由连接$\tilde{x}_1(n)$的两个周期得到。即:

$$\tilde{x}_3(n)=[\tilde{x}_1(n),\tilde{x}_1(n)]_{\text{PERIODIC}}$$

显然,$\tilde{x}_3(n)$与习题 3 中的 $\tilde{x}_2(n)$不同,尽管它们的周期均为 100。

a. 求出 $\tilde{x}_3(n)$的 DFS $\tilde{X}_3(k)$,画出它的幅度和角度样本曲线.

b. 周期加倍后对 DFS 有什么影响?

c. 推广上述结果,把周期扩至 M 倍,证明如果

$$\tilde{x}_M(n)=[\underbrace{\tilde{x}_1(n),\ldots,\tilde{x}_1(n)}_{M\text{次}}]_{\text{PERIODIC}}$$

则

$$\begin{aligned}\tilde{X}_M(Mk)&=M\tilde{X}_1(k),\ k=0,1,\ldots,N-1\\ \tilde{X}_M(k)&=0,\qquad k\neq 0,M,\ldots,MN\end{aligned}$$

P5.5 设 $X(e^{j\omega})$为 10 点序列 $x(n)=\{2,5,3,-4,-2,6,0,-3,-3,2\}$的 DTFT

a.设

$$y_1(n)=\overset{3\text{-点}}{\text{IDFS}}[X(e^{j0}),X(e^{j2\pi/3}),X(e^{j4\pi13})]$$

用频率采样定理求出 $y_1(n)$。用 MATLAB 验证你的结果。

b.设

$$y_2(n)=\overset{20\text{-点}}{\text{IDFS}}[X(e^{j0}),X(e^{j2\pi/20}),X(e^{j4\pi/3}),\ldots,X(e^{j2\pi(19)/20})]$$

用频率采样定理求出 $y_2(n)$。用 MATLAB 验证你的结果。

P5.6 12 点序列 $X(n)$定义为:$X(n)=\{1,2,3,4,5,6,6,5,4,3,2,1\}$

a.求出 $x(n)$的 DFT $X(k)$,画出它的幅度和相位曲线(使用 stem 函数)。

b.用 MATLAB 画出 $x(n)$的 DTFT $X(e^{j\omega})$的幅度和相位曲线。

c.验证 a 中的 DFT 是 $X(e^{j\omega})$的采样。采用 hold 函数把两图放在一幅图里,会有助于解此题。

d.有无可能从 DFT $X(k)$重构 DTFT $X(e^{j\omega})$? 如果可能,给出重构所需要的内插公式;若不可能,说明不能重构的理由。

P5.7 以 DFT 作计算工具,画出下列序列的 DTFT 幅度。设定合理的长度 N,使所作的图有意义。

a. $x_1(n)=2\cos(0.2\pi n)[u(n)-u(n-10)]$

b. $x_2(n)=\sin(0.45\pi n)\sin(0.55\pi n),\ 0\leqslant n\leqslant 50$

c. $x_3(n)=3(2)^n,\ -10\leqslant n\leqslant 10$

d. $x_4(n)=(-0.5)^n,\ -10\leqslant n\leqslant 10$

e. $x_5(n)=5(0.9e^{j\pi/4})^n u(n)$

P5.8 设 $H(e^{j\omega})$是因果的、离散时间、实 LTI 系统。

a.如果

$$\mathrm{Re}\{H(e^{j\omega})\}=\sum_{k=0}^{5}(0.5)^k\cos(k\omega)$$

解析地求出脉冲响应 $h(n)$,用 IDFT 作为计算工具,验证你的答案。请明智地选择长度 N。

b.如果

$$\mathrm{Im}\{H(e^{j\omega})\}=\sum_{l=0}^{5}2l\sin(l\omega)\quad 并且\quad \int_{-\pi}^{\pi}H(e^{j\omega})d\omega=0$$

解析地求出脉冲响应 $h(n)$,用 IDFT 作为计算工具,验证你的答案。请明智地选择长度 N。

P5.9 设 $X(k)$表示 N 点序列 $x(n)$的 N 点 DFT,DFT $X(k)$本身也为 N 点序列。

a.如果计算 $X(k)$的 DFT,得到另一个 N 点序列 $x_1(n)$,证明

$$x_1(n)=Nx((n))_N,\ 0\leqslant n\leqslant N-1$$

b.利用上述性质,设计一个 MATLAB 函数实现 N 点循环折叠运算 $x_2(n)=x_1((-n))_N$,该函数格式应为:

```
x2 = circfold(x1,N)
% 利用 DFT 进行的循环折叠
% x2 = circfold(x1,N)
% x2 = 循环折叠输出序列
% 长度 < = N 的输入序列
% N = 循环缓冲器长度
```

c.求出下列序列的循环折叠:$X_1(n)=\{1,2,3,4,5,6,6,5,4,3,2,1\}$

P5.10 用下列关系式把 N 点复序列分解成 N 点偶序列和奇序列

$$x_{ec}(n) \triangleq \frac{1}{2}[x(n) + x^*((-n))_N]$$

$$x_{oc}(n) \triangleq \frac{1}{2}[x(n) + x^*((-n))_N]$$

则

$$\text{DFT}[x_{ec}(n)] = \text{Re}[X(k)] = \text{Re}[X((-k))_N]$$
$$\text{DFT}[x_{oc}(n)] = \text{Im}[X(k)] = \text{Im}[X((-k))_N]$$

a.解析地证明上述性质。

b.修正本章开发的 circevod 函数,使它能用于复序列。

c.用下列序列验证上述对称性和你的 MATLAB 函数:

$$x(n) = (0.9e^{j\pi/3})^n[u(n) - u(n-20)]$$

P5.11 实序列 $x(n)$的 8 点 DFT 的前 5 个值为:{0.25,0.125 − j0.3,0,0.125 − j0.06, 0.5}

利用 DFT 性质,求出下列每一个序列的 DFT。

a. $x_1(n) = x((2-n))_8$

b. $x_2(n) = x((n+5))_{10}$

c. $x_3(n) = x^2(n)$

d. $x_4(n) = x(n)⑧x((-n))_8$

e. $x_5(n) = x(n)e^{j\pi n/4}$

P5.12 如果 $X(k)$是 N 点复序列 $x(n) = x_R(n) + jx_I(n)$的 DFT,其中 $X_R(n)$和 $X_I(n)$是 $X(n)$的实部和虚部,则

$$X_R(k) \triangleq \text{DFT}[x_R(n)] = X_{ec}(k)$$
$$X_I(k) \triangleq \text{DFT}[x_I(n)] = X_{oc}(k)$$

其中 $X_{ec}(k)$和 $X_{oc}(k)$是习题 5.10 中定义的 $X(k)$的循环偶分量和循环奇分量。

a.解析地证明上述性质。

b.利用此性质可通过一个 N 点 DFT 运算得到两个 N 点实序列的 DFT。明确地说,设 $x_1(n)$和 $x_2(n)$为两个 N 点序列。则可形成一个复序列:

$$x(n) = x_1(n) + jx_2(n),$$

利用上述性质,以下面的格式开发一个 MATLAB 函数来实现此方法

```
function [x1,x2] = real2dft(x1,x2,N)
```

```
% 两个实序列的 DFTs
% [X1,X2] = real2dft(x1,x2,N)
% X1 = x1 的 n 点的 DFT
% X2 = x2 的 n 点的 DFT
% x1 = 长度 < = N 的序列
% x2 = 长度 < = N 的序列
%   N = DFT 的长度
```

c.计算下面两个序列的 DFT。

$$x(n)=\cos(0.25\pi n),\ x(n)=\sin(0.75\pi n);\ 0\leqslant n\leqslant 63$$

P5.13 采用频域方法,开发一个 MATLAB 函数,在给定 N_1 点序列 $x(n)$时(其中 $N_1\leqslant N$),求出循环移位 $x((n-m))_N$,该程序应有下面的格式:

```
function y = cirshftf(x,m,N)
%
% function y = cirshftf(x,m,N)
%
% 长度为 N 的 x 序列:(频域)作 m 个样本的循环移位
% ————————————————————————————
% y : 包含循环移位的输出序列
% x : 长度 < = N 的输入序列
% m : 移位样本数
% N : 循环缓冲器长度
%
% 方法:y(n) = idft(dft(x(n)) * WN^(mk))
% 如果 m 是个标量,则 y 是一个序列(行向量)
% 如果 m 是个向量,则 y 是个矩阵,每一行是一个循环移位
% x 相应于向量 m 中的各项
% M 和 x 不能为矩阵
```

用下面的序列: $x_1(n)=11-n,\ 0\leqslant n\leqslant 10$ 验证你的函数,其中 $m=10, N=15$。

P5.14 利用 DFT 的分析和综合等式,证明:

$$\sum_{n=0}^{N-1}|x(n)|^2=\frac{1}{N}\sum_{k=0}^{N-1}|X(k)|^2$$

这就是通常所说的 DFT Parseval 定理,对习题 5.9 中的序列,用 MATLAB 验证此定理。

P5.15 采用频域方法,开发一个 MATLAB 函数实现两序列间循环卷积的运算,其格式

应为：

```
function x3 = circonvf(x1,x2,N)
% 频域循环卷积
% x3 = circonvf(x1,x2,N)
% x3 = 长度为 N 的卷积结果
% x1 = 长度 <= N 的输入序列
% x2 = 长度 <= N 的输入序列
% N = 循环缓冲器的长度
```

P5.16 本章中开发的 circonvt 函数用矩阵-矢量乘法实现循环卷积。相应于循环移位的矩阵$\{x((n-m))_N;\ 0\leqslant n\leqslant N-1\}$具有有趣的结构，此矩阵叫做循环矩阵，它是第二章中介绍的 Toeplitz 矩阵的特例。

a.考虑例 5.13 中所给序列，将 $x_1(n)$表示成列向量 x_1，$x_2((n-m))_N$ 表示成矩阵 X_2 形式，它的各行相应于 $n=0,1,2,3$.试说明矩阵 X_2 的特征，它能完全由第一行(或列)表征吗？

b. 用 X_2x_1 求出循环卷积。并验证你的结果。

P5.17 用习题 5.12 中的 cirshftf 函数，开发一个 MATLAB 函数来构造给定 N 点序列 $x(n)$的循环矩阵 $\boldsymbol{C}$，该函数应具有以下格式：

```
function[C] = circulnt(x,N)
% 从一 N 点序列得到的循环矩阵
% [C] = circulnt(x,N)
% C = 维数为 N×N 的循环矩阵
% x = 长度 < = N 的序列
% N = 循环矩阵的维数
```

用此函数修正本章中讨论的实现循环卷积的函数 circonvt，以取消 for...end 环。用习题 5.16 中的序列验证你的函数。

P5.18 计算下列序列的 N 点循环卷积。

a. $x_1(n)=\{1,1,1,1\},x_2(n)=\cos(\pi n/4)R_N(n);\ N=8$

b. $x_1(n)=\cos(2\pi n/N)R_N(n),x_2(n)=\sin(2\pi n/N)R_N(n);\ N=32$

c. $x_1(n)=(0.8)^nR_N(n),x_2(n)=(-0.8)^nR_N(n);\ N=20$

d. $x_1(n)=nR_N(n),x_2(n)=(N-n)R_N(n);\ N=10$

e. $x_1(n)=\{1,-1,1,-1\},x_2(n)=\{1,0,-1,0\};\ N=4$

P5.19 对于下列序列计算：(ⅰ)N 点循环卷积 $x_3(n)=x_1(n)\,Ⓝ\,x_2(n)$；(ⅱ)线性卷积 $x_4(n)=x_1(n)*x_2(n)$；(ⅲ)误差序列 $e(n)=x_3(n)-x_4(n)$。

a. $x_1(n)=\{1,1,1,1\},x_2(n)=\cos(\pi n/4)R_6(n);\ N=8$

b. $x_1(n)=\cos(2\pi n/N)R_{16}(n),x_2(n)=\sin(2\pi n/N)R_{16}(n);\ N=32$

c. $x_1(n)=(0.8)^nR_{10}(n),x_2(n)=(-0.8)^nR_{10}(n);\ N=15$

d. $x_1(n)=nR_{10}(n),x_2(n)=(N-n)R_{10}(n);\ N=10$

e. $x_1(n)=\{1,-1,1,-1\}$, $x_2(n)=\{1,0,-1,0\}$; $N=5$

在每种情况下验证 $e(n)=x_4(n+N)$。

P5.20 分段卷积的重叠相加法是重叠保留法的一种替代方法。设 $x(n)$是长度为 ML 的长序列,其中 $M,L>>1$,把 $x(n)$分成 M 段$\{x_m(n), m=1......M\}$,每段长度为 L。

$$x_m(n)=\begin{cases}x(n), mM\leqslant n\leqslant(m+1)M-1\\0, \quad 其它\end{cases} \qquad 则 \quad x(n)=\sum_{m=0}^{M-1}x_m(n)$$

设 $h(n)$为 L 点脉冲响应;则

$$y(n)=x(n)*h(n)=\sum_{m=0}^{M-1}x_m(n)*h(n)=\sum_{m=0}^{M-1}y_m(n);\ y_m(n)\triangleq x_m(n)*h(n)$$

显然,$y_m(n)$是$(2L-1)$点序列。在这种方法中,需要保存中间卷积结果,在相加之前进行恰当的重叠,形成 $y(n)$。为在运算中使用 DFT,必须选择 $N\geqslant(2L-1)$。

a.利用循环卷积,开发一个 MATLAB 函数实现重叠相加法,其格式应为:

```
function [y] = ovrlpadd(x,h,N)
% 用重叠相加法作分段卷积
% [y] = ovrlpsav(x,h,N)
%
% y = 输出序列
% x = 输入序列
% h = 脉冲响应
% N = 段长 > = 2 * length(h) - 1
```

b.在上述函数中采用基 2 FFT,以得到一个高速重叠相加分段卷积的程序。记住选择 $N=2^\nu$。

c.用下列两个序列验证你的函数。

$$x(n)=\cos(\pi n/500)R_{4000}(n),\ h(n)=\{1,-1,1,-1\}$$

P5.21 给定序列 $x_1(n)$和 $x_2(n)$为:

$$x_1(n)=\{2,1,1,2\},\ x_2(n)=\{1,-1,-1,1\}$$

a.计算 $N=4,7$ 和 8 时的循环卷积 $x_1(n)ⓃX_2(n)$。

b.计算线性卷积 $x_1(n)*x_2(n)$。

c.利用计算结果,求出在 N 点区间上线性卷积和循环卷积相等所需要的最小 N 值。

d.不要实际计算卷积,说明你怎样得到 c 中的结果。

P5.22 设

$$x(n)=\begin{cases}A\cos(2\pi ln/N), & 0\leqslant n\leqslant N-1\\ 0, & \text{其它}\end{cases}=A\cos(2\pi ln/N)R_N(n)$$

其中 l 为整数。注意 $x(n)$ 在 N 个样本中包含余弦的 l 个周期，这是一个由窗口截取的无泄漏的余弦序列。

a. 证明它的 DFT $X(k)$ 为实序列：

$$X(k)=\frac{AN}{2}\delta(k-l)+\frac{AN}{2}\delta(k-N+l);\ 0\leqslant k\leqslant(N-1), 0<l<N$$

b. 证明若 $l=0$，则 DFT $X(k)$ 由下式给出：

$$X(k)=AN\delta(k);\ 0\leqslant k\leqslant(N-1)$$

c. 解释当 $l<0$ 或 $l>N$ 时，如何修正上式。

d. 用下面的序列验证 a、b、c 中的结果，利用 stem 函数画出 DFT 序列的实部。

(ⅰ) $x_1(n)=3\cos(0.04\pi n)R_{200}(n)$

(ⅱ) $x_2(n)=5R_{50}(n)$

(ⅲ) $x_3(n)=[1+2\cos(0.5\pi n)+\cos(\pi n)]R_{100}(n)$

(ⅳ) $x_4(n)=\cos(25\pi n/16)R_{64}(n)$

(ⅴ) $x_5(n)=[4\cos(0.1\pi n)-3\cos(1.9\pi n)]R_N(n)$

P5.23 设 $x(n)=A\cos(\omega_0 n)R_N(n)$，其中 ω_0 为实数。

a. 利用 DFT 性质，证明 $X(k)$ 的实部和虚部为：

$$X(k)=X_R(k)+jX_I(k)$$

$$X_R(k)=(A/2)\cos\left[\frac{\pi(N-1)}{N}(k-f_0N)\right]\frac{\sin[\pi(k-f_0N)]}{\sin[\pi(k-f_0N)/N]}$$

$$+(A/2)\cos\left[\frac{\pi(N-1)}{N}(k+f_0N)\right]\frac{\sin[\pi(k-N+f_0N)]}{\sin[\pi(k-N+f_0N)/N]}$$

$$X_I(k)=-(A/2)\sin\left[\frac{\pi(N-1)}{N}(k-f_0N)\right]\frac{\sin[\pi(k-f_0N)]}{\sin[\pi(k-f_0N)/N]}$$

$$-(A/2)\sin\left[\frac{\pi(N-1)}{N}(k+f_0N)\right]\frac{\sin[\pi(k-N+f_0N)]}{\sin[\pi(k-N+f_0N)/N]}$$

b. 上述结果说明，余弦信号的初始频率 ω_0 泄漏进形成时间有限序列的谐波的其他频率里，因此把它叫做余弦泄漏性质。它是由于在非整数周期上对带宽有限的周期余弦信号进行采样产生的必然结果。用 $x(n)$ 的周期延伸 $\tilde{X}(n)$ 和习题 5.22a 中的结论，解释此结果。

c. 用 $x(n)=\cos(5\pi n/99)R_{200}(n)$ 验证泄漏特性，利用 stem 函数画出 DFT 序列的实部和虚部。

P5.24 设

$$x(n)=\begin{cases}A\sin(2\pi ln/N), & 0\leqslant n\leqslant N-1\\ 0, & \text{其它}\end{cases}=A\sin(2\pi ln/N)R_N(n)$$

其中 l 为整数。注意 $x(n)$在 N 个样本中包含正弦的 l 个周期,这是一个由窗口截取的无泄漏的正弦序列。

a.证明它的 DFT $X(k)$为纯虚序列:

$$X(k)=\frac{AN}{2j}\delta(k-l)-\frac{AN}{2j}\delta(k-N+l);\ 0\leqslant k\leqslant(N-1),0<l<N$$

b.证明若 $l=0$,则 DFT $X(k)$由下式给出:

$$X(k)=0;\quad 0\leqslant k\leqslant(N-1)$$

c.解释当 $l<0$ 或 $l>N$ 时,如何修正上式。

d.用下面的序列验证 a、b、c 中的结果,利用 stem 函数画出 DFT 序列的虚部。

(ⅰ) $x_1(n)=3\sin(0.04\pi n)R_{200}(n)$

(ⅱ) $x_2(n)=5\sin 10\pi nR_{50}(n)$

(ⅲ) $x_3(n)=[2\sin(0.5\pi n)+\sin(\pi n)]R_{100}(n)$

(ⅳ) $x_4(n)=\sin(25\pi n/16)R_{64}(n)$

(ⅴ) $x_5(n)=[4\sin(0.1\pi n)-3\sin(1.9\pi n)]R_N(n)$

P5.25 设 $x(n)=A\sin(\omega_0 n)R_N(n)$,其中 ω_0 为实数。

a. 利用 DFT 性质,证明 $X(k)$的实部和虚部为:

$$\begin{aligned}
X(k)&=X_R(k)+jX_I(k)\\
X_R(k)&=(A/2)\sin\left[\frac{\pi(N-1)}{N}(k-f_0N)\right]\frac{\sin[\pi(k-f_0N)]}{\sin[\pi(k-f_0N)/N]}\\
&\quad-(A/2)\sin\left[\frac{\pi(N-1)}{N}(k+f_0N)\right]\frac{\sin[\pi(k-N+f_0N)]}{\sin[\pi(k-N+f_0N)/N]}\\
X_I(k)&=-(A/2)\cos\left[\frac{\pi(N-1)}{N}(k-f_0N)\right]\frac{\sin[\pi(k-f_0N)]}{\sin[\pi(k-f_0N)/N]}\\
&\quad+(A/2)\cos\left[\frac{\pi(N-1)}{N}(k+f_0N)\right]\frac{\sin[\pi(k-N+f_0N)]}{\sin[\pi(k-N+f_0N)/N]}
\end{aligned}$$

b.上述结果为正弦泄漏性质。用 $x(n)$的周期延伸 $\tilde{X}(n)$和习题 5.24 a 中的结论解释此结果。

c. 用 $x(n)=\sin(5\pi n/99)R_{200}(n)$验证泄漏特性,利用 stem 函数画出 DFT 序列的实部和虚部。

P5.26 对模拟信号 $x_a(t)=2\sin(4\pi t)+5\cos(8\pi t)$在 $t=0.01n,n=0,1,\ldots N-1$ 上采样,得到 N 点序列 $x(n)$,用 N 点 DFT 得到对 $x_a(t)$幅度谱的估计。

a.从下面值中,选择一个能提供最精确的 $x_a(t)$的幅度谱的 N,画出 DFT 幅度谱 $|X(k)|$ 的实部和虚部。(1) $N=40$ (2) $N=50$ (3) $N=60$

b. 从下面值中，选择一个能提供最小 $x_a(t)$ 的幅度谱泄漏量的 N，画出 DFT 幅度谱 $|X(k)|$ 的实部和虚部。(1) $N=90$ (2) $N=95$ (3) $N=99$

P5.27 利用(5.49)，求出并画出 $N=8$ 点，基-2 频率抽选 FFT 算法的信号流图，根据流图，确定下面序列的 DFT。

$$x(n)=\cos(\pi n/2),\ 0\leqslant n\leqslant 7$$

P5.28 利用(5.49)，求出并画出 $N=16$ 点，基-4 时间抽选 FFT 算法的信号流图，根据流图，确定下面序列的 DFT。

$$x(n)=\cos(\pi n/2),\ 0\leqslant n\leqslant 15$$

P5.29 设 $x(n)=\cos(\pi n/99)$，$0\leqslant n\leqslant(N-1)$ 为 N 点序列。选择 $N=4^{\nu}$，求出 $\nu=5,6,\ldots 10$ 的 MATLAB 执行时间。验证这些时间与 $N\log_4 N$ 成正比。

第六章　数字滤波器结构

在前面几章中，我们研究了时域及频域中离散系统的有关理论，现在将把它应用于数字信号处理。为了处理信号，必须设计和实现滤波器（在一些文章中，也叫做频谱分析装置）系统。滤波器的设计结果受滤波器的类型（即，IIR 或 FIR）和其实现形式（结构）的影响。因此在讨论设计结果之前，先考虑在实际中怎样实现这些滤波器，这一点是很重要的，因为不同的滤波器结构采用不同的设计策略。

正如前面所讨论的，IIR 滤波器的特征是具有无限持续时间脉冲响应。其中一些脉冲响应可以用有理系统函数或差分方程来建模，这样的滤波器叫做自回归滑动平均（ARMA），更一般地，称为递归滤波器，而不能这样建模的 IIR 滤波器叫做非递归滤波器。在 DSP 中，IIR 滤波器一般是指递归的，这是由于它们能被经济地实现。而且，ARMA 滤波器包括 FIR 滤波器的滑动平均滤波器。出于设计和实现目的，我们把 FIR 滤波器和 IIR 滤波器分开考虑。

本章首先给出用来描述滤波器结构的基本结构框图，随之简要叙述 IIR、FIR 和格型滤波器结构，并相应地提供实现这些结构的 MATLAB 函数。

基本元件

既然滤波器为 LTI 系统，则我们需要用下面三种元件描述数字滤波器结构，这些元件如图 6.1 所示。

- **加法器**：此元件有两个输入和一个输出，如图 6.1(a)所示。注意三个或多个信号的相加由相连的两输入加法器实现。
- **乘法器（增益）**：这是一个单入单出元件，如图 6.1(b)所示。由于乘 1 时易于理解，因此这里不明确给出示意图。
- **延迟元件（移位或记忆）**：此元件把通过它的信号延迟一个样本，如图 6.1(c)所示。它由移位寄存器实现。

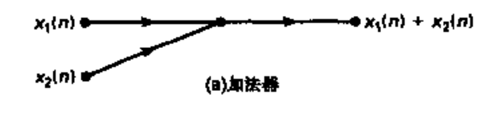


图 6.1　三个基本元件

利用这些基本元件，我们可描述 IIR 和 FIR 的各种结构，为了实现这些结构，需要对多项式进行运算，因此 MATLAB 是一个很方便的工具。

IIR 滤波器结构

一个 IIR 滤波器的系统函数为：

$$H(z)=\frac{B(z)}{A(z)}=\frac{\sum_{n=0}^{M}b_nz^{-n}}{\sum_{n=0}^{N}a_nz^{-n}}=\frac{b_0+b_1z^{-1}+\cdots+b_Mz^{-M}}{1+a_1z^{-1}+\cdots+a_Nz^{-N}};a_0=1 \tag{6.1}$$

其中 b_n,a_n 是滤波器系数。不失一般性,假设 $a_0=1$。如果 $a_N\neq0$,这时 IIR 滤波器阶数为 N。IIR 滤波器的差分方程为:

$$y(n)=\sum_{m=0}^{M}b_mx(n-m)-\sum_{m=1}^{N}a_my(n-m) \tag{6.2}$$

实现 IIR 滤波器的三种不同结构为:

- **直接形式**:在此形式中,差分方程(6.2)以给定的形式直接实现。滤波器分为两个部分,滑动平均部分和递归部分(或者,分子和分母部分)。因此,这种实现有两种形式:直接Ⅰ型和直接Ⅱ型结构。
- **级联形式**:在此形式中,把等式(6.1)中的系统函数 $H(z)$因式分解成二阶子系统,每个子系统叫做二阶环节。因此系统函数可表示成这些二阶环节的乘积形式。每个二阶环节以直接形式实现,整个系统函数由二阶环节的级联实现。
- **并联形式**:这种形式与级联形式类似,但因式分解后,用部分分式展开把 $H(z)$表示成二阶子系统的和。每一子系统用直接形式实现,整个系统函数以子系统的并联网络实现。

在这节中,我们简要讨论这些形式。一般地讲,用系统函数的有理式(或直接形式结构)描述 IIR 滤波器。随后我们提供 MATLAB 函数把直接形式结构转换成级联和并联形式结构。

直接形式

正如它名字所揭示的,用延迟元件、乘法器和加法器,以给定的形式直接实现差分方程(6.2)。为了具体说明,设 $M=N=4$,那么差分方程为:

$$\begin{aligned}y(n)=&b_0x(n)+b_1x(n-1)+b_2x(n-2)+b_3x(n-3)+b_4x(n-4)\\&-a_1y(n-1)-a_2y(n-2)-a_3y(n-3)-a_4y(n-4)\end{aligned}$$

它的具体实现如图 6.2 所示。这种方框图叫做直接Ⅰ型结构。

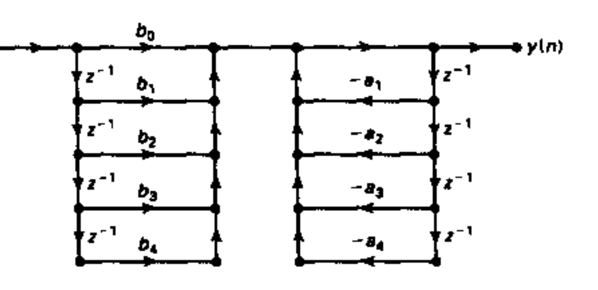


图 6.2　直接Ⅰ型结构

直接Ⅰ型结构独立地实现有理函数 $H(z)$的每一部分,然后把它们级联起来。分子部分是抽头延迟线,在它后面为分母部分,为反馈抽头延迟线。在此结构中存在两部分独立的延迟线,因此,需要 8 个延迟元件。我们可交换两部分的级联次序,使延迟元件的个数减少。现在,把两个延迟线并排,由一个单位增益分支连在一起,因此可拿掉其中的一个延迟线,这种缩减导致一种标准的结构,叫做直接Ⅱ型结构,如图 6.3 所示。注意,从输入输出的观点看,这两种直接形式

是等价的。但在内部，它们具有不同的信号。

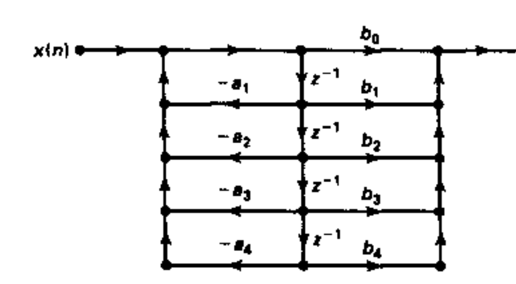


图 6.3 直接Ⅱ型结构

MATLAB 实现

在 MATLAB 中，直接形式结构由两个行矢量描述；b 包含{bn}系数，a 包含{an}系数。它由第二章讨论的 filter 函数实现。

级联形式

在这种形式中，系统函数 $H(z)$ 写成实系数二阶子系统的乘积形式。首先把分子、分母多项式的根解出，然后把每一对共轭复根或任意两个实根组合在一起，得到二阶子系统。在本章下面各节中，假设 N 为偶数，则：

$$
\begin{aligned}
H(z) &= \frac{b_0 + b_1 z^{-1} + \cdots + b_N z^{-N}}{1 + a_1 z^{-1} + \cdots + a_N z^{-N}} \\
&= b_0 \frac{1 + \dfrac{b_1}{b_0} z^{-1} + \cdots + \dfrac{b_N}{b_0} z^{-N}}{1 + a_1 z^{-1} + \cdots + a_N z^{-N}} \\
&= b_0 \prod_{k=1}^{K} \frac{1 + B_{k,1} z^{-1} + B_{k,2} z^{-2}}{1 + A_{k,1} z^{-1} + A_{k,2} z^{-2}}
\end{aligned}
\tag{6.3}
$$

其中 K 等于 $N/2$，$B_{k,1}$、$B_{k,2}$、$A_{k,1}$、$A_{k,2}$ 为实数，表示二阶子系统的系数。二阶子系统

$$
H_k(z) = \frac{Y_{k+1}(z)}{Y_k(z)} = \frac{1 + B_{k,1} z^{-1} + B_{k,2} z^{-2}}{1 + A_{k,1} z^{-1} + A_{k,2} z^{-2}}; \quad k = 1, \ldots, K
$$

其中

$$
Y_1(z) = b_0 X(z); \quad Y_{K+1}(z) = Y(z)
$$

叫做第 k 个双二阶环节(biquad)。它的输入是第 $(k-1)$ 个双二阶环节的输出，同时第 k 个双二阶环节的输出为第 $(k+1)$ 个双二阶环节的输入。每一个双二阶环节 $H_k(z)$ 可用图 6.4 所示的直接Ⅱ型实现。整个滤波器由双二阶环节的级联形式实现。

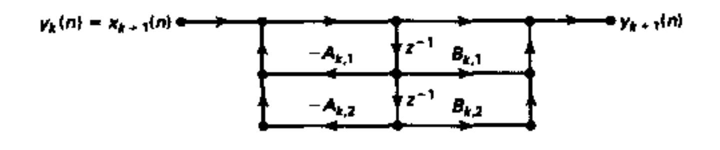


图 6.4 双二阶环节结构

作为一个例子，假设 $N=4$。图 6.5 给出此 4 阶 IIR 滤波器的级联形式结构。

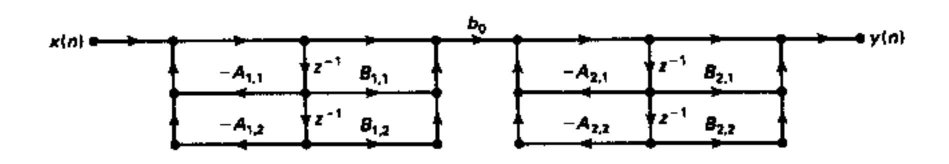


图 6.5　$N=4$ 的级联形式结构

MATLAB 实现

已知直接形式滤波器的系数 $\{b_n\}$ 和 $\{a_n\}$，我们必须得到系数 b_0，$\{B_{k,i}\}$ 和 $\{A_{k,i}\}$，这由下面的 dir2cas 函数完成。

```
function [b0,B,A] = dir2cas(b,a);
% 直接型到级联型的形式转换(复数对型)
% ─────────────────────────────
% [b0,B,A] = dir2cas(b,a)
% b = 直接型的分子多项式系数
% a = 直接型的分母多项式系数
% b0 = 增益系数
% B = 包含各 bk 的 K 乘 3 维实系数矩阵
% A = 包含各 ak 的 K 乘 3 维实系数矩阵

% 计算增益系数 b0
b0 = b(1); b = b/b0;
a0 = a(1); a = a/a0;
b0 = b0/a0;
%
M = length(b); N = length(a);
if N > M
    b = [b zeros(1,N - M)];
elseif M > N
    a = [a zeros(1,M - N)]; N = M;
else
    NM = 0;
end
%
K = floor(N/2); B = zeros(K,3); A = zeros(K,3);
if K * 2 = = N;
    b = [b 0];
    a = [a 0];
end
```

```
%
broots = cplxpair(roots(b));
aroots = cplxpair(roots(a));
for i = 1:2:2 * K
Brow = broots(i:1:i + 1,:);
Brow = real(poly(Brow));
B(fix((i + 1)/2),:) = Brow;
Arow = aroots(i:1:i + 1,:);
Arow = real(poly(Arow));
A(fix((i + 1)/2),:) = Arow;
end
```

上述函数把矢量 b 和 a 转换成 $K \times 3$ 维矩阵 B 和 A。首先计算 b_0，它等于 b_0/a_0，(假设 $a_0 \neq 1$)，然后通过给较短的矢量填零，使矢量 b 和 a 一样长。这就保证每一个双二阶环节的分子和分母均不为零。接着计算多项式 $B(z)$ 和 $A(z)$ 的根。用 cplxpair 函数把这些根以共轭复根对的次序排列。然后用 poly 函数把每一对根再转换成二阶分子或分母多项式。

级联形式由下面的 casfiltr 函数实现。它把每个二阶环节的系数存在矩阵 B 和 A 中，放在一个循环里，而在此循环中采用 filter 函数。输入被乘以 b_0，每个滤波器的输出作为下一级滤波器的输入，最后一个滤波器的输出即为总的输出。

```
function y = casfiltr(b0,B,A,x);
% IIR 和 FIR 滤波器的级联型的实现
% ——————————————————————
% y = casfiltr(b0,B,A,x);
% y = 输出序列
% b0 = 级联型的增益系数
% B = 包含各 bk 的 K 乘 3 维实系数矩阵
% A = 包含各 ak 的 K 乘 3 维实系数矩阵
% x = 输入序列
%
[K,L] = size(B);
N = length(x);
w = zeros(K + 1,N);
w(1,:) = x;
for i = 1:1:K
    w(i + 1,:) = filter(B(i,:),A(i,:),w(i,:));
end
y = b0 * w(K + 1,:);
```

下面的 MATLAB 函数 cas2dir 把级联形式转换成直接形式。这是一个包含几次多项式乘

法的简单运算。为达到这个目的,在一个 K 次的循环中使用 conv 函数。

```
function [b,a] = cas2dir(b0,B,A);
% 级联型到直接型的形式转换
% ————————————————
% [b,a] = cas2dir(b0,B,A)
% b = 直接型的分子多项式系数
% a = 直接型的分母多项式系数
% b0 = 增益系数
% B = 包含各 bk 的 K 乘 3 维实系数矩阵
% A = 包含各 ak 的 K 乘 3 维实系数矩阵
%
[K,L] = size(B);
b = [1];
a = [1];
for i = 1:1:K
    b = conv(b,B(i,:));
    a = conv(a,A(i,:));
end
b = b * b0;
```

例 6.1 一个滤波器由下面的差分方程描述:

$$
\begin{aligned}
&16y(n)+12y(n-1)+2y(n-2)-4y(n-3)-y(n-4)\\
&\quad =x(n)-3x(n-1)+11x(n-2)-27x(n-3)+18x(n-4)
\end{aligned}
$$

求出它的级联形式结构。

解:MATLAB 程序

```
>> b = [1, -3, 11, -27, 18];
>> a = [16, 12, 2, -4, -1];
>> [b0,B,A] = dir2cas(b,a)
b0 = 0.0625
B =
    1.0000    0.0000    9.0000
    1.0000   -3.0000    2.0000
A =
    1.0000    1.0000    0.5000
    1.0000   -0.2500   -0.1250
```

最后所得的结构如图 6.6 所示。为了检查级联结构是否正确，用两种形式计算脉冲响应的前 8 个样本。

```
>> delta=impseq(0,0,7);
delta =
      1    0    0    0    0    0    0    0
>> format long
>> hcas=casfiltr(b0,B,A,delta)
hcas =
  Columns 1 through 4
  0.06250000000000  -0.23437500000000  0.85546875000000  -2.28417968750000
  Columns 5 through 8
  2.67651367187500  -1.52264404296875  0.28984069824219  0.49931716918945
>> hdir=filter(b,a,delta)
hdir =
  Columns 1 through 4
  0.06250000000000  -0.23437500000000  0.85546875000000  -2.28417968750000
  Columns 5 through 8
  2.67651367187500  -1.52264404296875  0.28984069824219  0.49931716918945
```

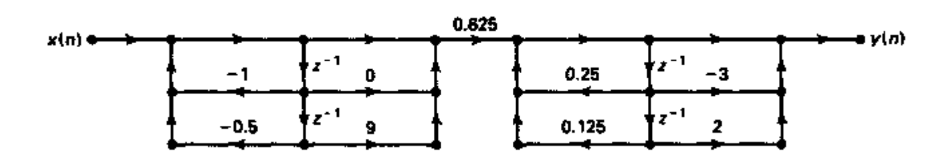


图 6.6　例 6.1 中的级联结构

并联形式

在这种形式中，系统函数 $H(z)$ 用部分分式展开式(PFE)写成二阶子系统的和的形式。

$$H(z)=\frac{B(z)}{A(z)}=\frac{b_0+b_1z^{-1}+\cdots+b_Mz^{-M}}{1+a_1z^{-1}+\cdots+a_Nz^{-N}} \tag{6.4}$$

$$=\frac{\hat{b}_0+\hat{b}_1z^{-1}+\cdots+\hat{b}_{N-1}z^{1-N}}{1+a_1z^{-1}+\cdots+a_Nz^{-N}}+\underbrace{\sum_{0}^{M-N}C_kz^{-k}}_{\text{only if } M\geq N}$$

$$=\sum_{k=1}^{K}\frac{B_{k,0}+B_{k,1}z^{-1}}{1+A_{k,1}z^{-1}+A_{k,2}z^{-2}}+\underbrace{\sum_{0}^{M-N}C_kz^{-k}}_{\text{only if } M\geq N}$$

其中 K 等于 $N/2$，$B_{k,0}$、$B_{k,1}$、$A_{k,1}$、$A_{k,2}$ 为实数，表示二阶子系统的系数。二阶子系统

$$H_k(z)=\frac{Y_{k+1}(z)}{Y_k(z)}=\frac{B_{k,0}+B_{k,1}z^{-1}}{1+A_{k,1}z^{-1}+A_{k,2}z^{-2}};\ k=1,\ldots,K$$

且

$$Y_k(z)=H_k(z)X(z),\ Y(z)=\sum Y_k(z),\ M<N$$

叫做第 k 阶有理双二阶环节。滤波器输入对所有的双二阶环节均有效，同时若 $M \geqslant N$(FIR部分)，它也是多项式部分的输入。这些环节的和形成滤波器的输出。每一个双二阶环节 $H_k(z)$可用直接Ⅱ型实现。由于取决于各子环节的和，因此可建造一个并联结构实现$H(z)$。作为一个例子，使 $M=N=4$。图 6.7 给出此 4 阶 IIR 滤波器的并联形式的结构。

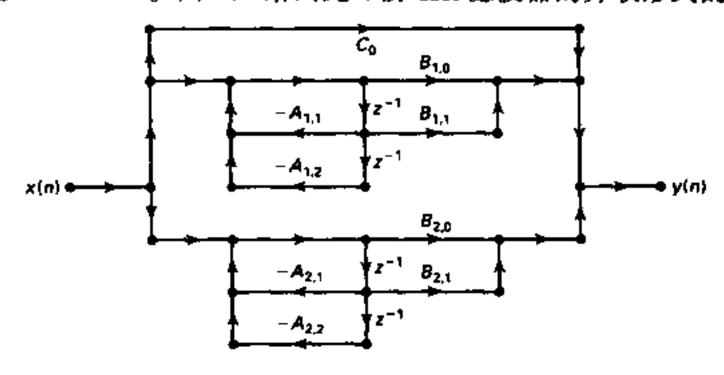


图 6.7　$N=4$ 的并联形式结构

MATLAB 实现

下面给出的 dir2par* 函数把直接型系数 $\{b_n\}$ 和 $\{a_n\}$ 转换成并联形式系数 $\{B_{k,i}\}$ 和 $\{A_{k,i}\}$。

```
function [C,B,A] = dir2par(b,a);
% 直接型到并联型的转换
% ——————————————
% [C,B,A] = dir2par(b,a)
% C = 当 length(b) > = length(a)时的多项式部分
% B = 包含各 bk 的 K 乘 2 维实系数矩阵
% A = 包含各 ak 的 K 乘 3 维实系数矩阵
% b = 直接型的分子多项式系数
% a = 直接型的分母多项式系数
%
M = length(b); N = length(a);
```

* ：若用旧版的 MATLAB 信号处理工具箱，此函数应作修改。参阅本书所附软盘中工具箱 PWS _ dspc 中的 dir2par 函数。

```
[r1,p1,C] = residuez(b,a);
p = cplxpair(p1,10000000 * eps);
I = cplxcomp(p1,p);
r = r1(I);

K = floor(N/2); B = zeros(K,2); A = zeros(K,3);
if K * 2 = = N; %N 为偶, A(z) 的次数为奇,有一个因子是一阶的
for i = 1:2:N - 2
Brow = r(i:1:i + 1,:);
Arow = p(i:1:i + 1,:);
[Brow,Arow] = residuez(Brow,Arow,[]);
B(fix((i + 1)/2),:) = real(Brow');
A(fix((i + 1)/2),:) = real(Arow');
end
[Brow,Arow] = residuez(r(N - 1),p(N - 1),[]);
B(K,:) = [real(Brow') 0]; A(K,:) = [real(Arow') 0];
else
      for i = 1:2:N - 1
Brow = r(i:1:i + 1,:);
Arow = p(i:1:i + 1,:);
[Brow,Arow] = residuez(Brow,Arow,[]);
B(fix((i + 1)/2),:) = real(Brow');
A(fix((i + 1)/2),:) = real(Arow');
end
end
```

dir2par 函数首先在 z 域里用 residuez 函数计算部分分式展开。我们需要把极点-留数对按复共轭极点-留数对、实极点-留数对的顺序进行排列,用 MATLAB 中的 cplxpair 函数可做到这一点,它可把复数数组分类为复共轭对。但由于连续两次调用此函数(一次为极点,一次为留数),不能保证极点和留数的互相对应。因此开发了一个新的 cplxcomp 函数,它把两个混乱的复数数组进行比较,返回一个数组的下标,用它重新给另一个数组排序。

```
function I = cplxcomp(p1,p2)
% I = cplxcomp(p1,p2)
% 比较两个包含同样标量元素但(可能)有不同下标的复数对
% 本程序必须用在 CPLXPAIR 程序之后以便重新排序频率极点向量
% 及其相应的留数向量:
%        p2 = cplxpair(p1)
%
I = [];
```

```
for j = 1:1:length(p2)
    for i = 1:1:length(p1)
if (abs(p1(i) - p2(j)) < 0.0001)
I = [I,i];
  end
    end
end
I = I';
```

得到极点-留数对后，dir2par 函数用 residuez 函数的逆形式计算双二阶环节的分子和分母。

在实现并联形式的 parfiltr 函数中使用这些并联形式的系数。parfiltr 函数把每个二阶环节的系数存在矩阵 B 和 A 中，并放在一个循环中，在此循环中，使用 filter 函数输入首先经过 FIRC 部分的滤波，其结果存在矩阵 w 的第一行中，随后，计算对相同输入下的所有二阶滤波器的输出，并把结果存在矩阵 w 的各行中，最后，把 w 中所有的列相加，产生最终的输出。

```
function y = parfiltr(C,B,A,x);
% IIR 滤波器的并联型实现
% ———————————————
% [y] = parfiltr(C,B,A,x);
% y = 输出序列
% C = 当 M >= N 时(FIR) 的多项式部分
% B = 包含各 bk 的 K 乘 2 维实系数矩阵
% A = 包含各 ak 的 K 乘 3 维实系数矩阵
% x = 输入序列
%
[K,L] = size(B);
N = length(x);
w = zeros(K+1,N);
w(1,:) = filter(C,1,x);
for i = 1:1:K
      w(i+1,:) = filter(B(i,:),A(i,:),x);
end
y = sum(w);
```

函数 par2dir 可用来从并联形式得到直接形式，它计算每一个双二阶环节的极点和留数，然后把它们组合成系统的极点和留数。再反调用 residuez 函数，计算分子和分母多项式。

```
function [b,a] = par2dir(C,B,A);
% 并联型到直接型的转换
% ———————————————
```

```
% [b,a] = par2dir(C,B,A)
% b = 直接型的分子多项式系数
% a = 直接型的分母多项式系数
% C = 并联型的多项式部分
% B = 包含各 bk 的 K 乘 2 维实系数矩阵
% A = 包含各 ak 的 K 乘 3 维实系数矩阵
%
[K,L] = size(A); R = []; P = [];

for i = 1:1:K
[r,p,k] = residuez(B(i,:),A(i,:));
R = [R;r]; P = [P;p];
end
[b,a] = residuez(R,P,C);
b = b(:)'; a = a(:)';
```

例 6.2 虑例 6.1 中的滤波器。

$$
\begin{aligned}
16y(n) + 12y(n-1) + 2y(n-2) - 4y(n-3) - y(n-4) \\
= x(n) - 3x(n-1) + 11x(n-2) - 27x(n-3) + 18x(n-4)
\end{aligned}
$$

求出它的并联形式。

解:MATLAB 程序

```
>> b=[1,-3,11,-27,18];
>> a=[16,12,2,-4,-1];
>>[C,B,A] = dir2par(b,a)
C =
  -18
B =
  10.0500   -3.9500
  28.1125   -13.362500
A =
  1.0000    1.0000    0.5000
  1.0000   -0.2500   -0.1250
```

最后所得的结构如图 6.8 所示。为了检查并联结构是否正确,用两种形式计算脉冲响应的前 8 个样本。

```
>> format long; delta = impseq(0,0,7);
```

```
>> hpar=parfiltr(C,B,A,delta)
hpar =
  Columns 1 through 4
  0.06250000000000  -0.23437500000001  0.85546875000000  -2.28417968750000
  Columns 5 through 8
  2.67651367187500  -1.52264404296875  0.28984069824219  0.49931716918945
>> hdir=filter(b,a,delta)
hdir =
  Columns 1 through 4
  0.06250000000000  -0.23437500000000  0.85546875000000  -2.28417968750000
  Columns 5 through 8
  2.67651367187500  -1.52264404296875  0.28984069824219  0.49931716918945
```

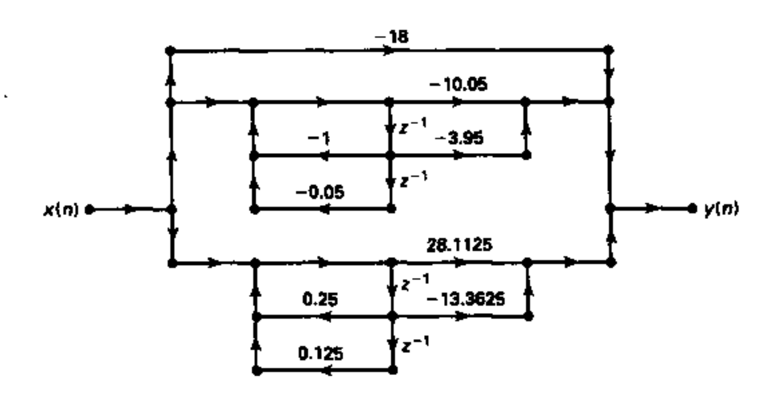


图 6.8 例 6.1 中的并联结构

例 6.3 如果一个滤波器结构中包括直接、级联、并联形式的组合，它用这三种形式表示的总特性是什么？考察如图 6.9 所示的方框图。

解：在这种结构中，包含两个并联子系统的级联形式。第一个并联子系统包含两个双二阶环节，而第二个子系统包含三个双二阶环节。我们得用 par2dir 函数把每个并联子系统转换成直接形式，然后把它们级联起来。最终的直接形式通过分别对相应的分子、分母多项式进行卷积得到。最终的级联和并联形式则可由直接形式导出。

```
>> C0=0; B1=[2,4;3,1]; A1=[1,1,0.9;1,0.4,-0.4];
>> B2=[0.5,0.7;1.5,2.5;0.8,1]; A2=[1,-1,0.8;1,0.5,0.5;1,0,-0.5];
>> [b1,a1]=par2dir(C0,B1,A1)
b1 =
    5.0000  8.8000  4.5000  -0.7000
a1 =
    1.0000  1.4000  0.9000  -0.0400  -0.3600
>> [b2,a2]=par2dir(C0,B2,A2)
b2 =
    2.8000  2.5500  -1.5600  2.0950  0.5700  -0.7750
```

```
a2 =
    1.0000  -0.5000  0.3000  0.1500  0.0000  0.0500  -0.2000
>> b=conv(b1,b2) %总的直接形式的分子
b =
  Columns 1 through 7
  14.0000  37.3900  27.2400  6.2620  12.4810  11.6605  -5.7215
  Columns 8 through 9
  -3.8865  0.5425
>> a=conv(a1,a2) %总的直接形式的分母
a =
  Columns 1 through 7
  1.0000  0.9000  0.5000  0.0800  0.1400  0.3530  -0.2440
  Columns 8 through 11
  -0.2890  -0.1820  -0.0100  0.0720
>> [b0,Bc,Ac]=dir2cas(b,a) 总的级联形式
b0 =
    14.0000
Bc =
    1.0000    1.8836    1.1328
    1.0000   -0.6915    0.6719
    1.0000    2.0776    0.8666
    1.0000         0         0
    1.0000   -0.5990    0.0588
Ac =
    1.0000    1.0000    0.9000
    1.0000    0.5000    0.5000
    1.0000   -1.0000    0.8000
    1.0000    1.5704    0.6105
    1.0000   -1.1704    0.3276
>> [C0,Bp,Ap]=dir2par(b,a)      % 总的并联形式
C0=[]
Bp =
  -20.4201   -1.6000
   24.1602    5.1448
    2.4570    3.3774
   -0.8101   -0.2382
    8.6129   -4.0439
Ap =
    1.0000    1.0000    0.9000
    1.0000    0.5000    0.5000
```

```
1.0000   -1.0000    0.8000
1.0000    1.5704    0.6105
1.0000   -1.1704    0.3276
```

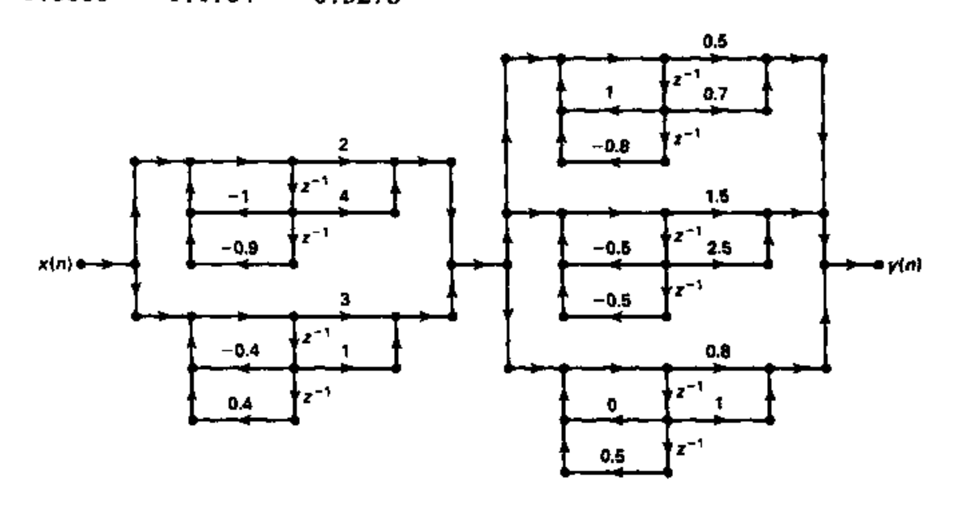


图 6.9　例 6.3 中的方框图

这个例子表明,利用本节开发的 MATLAB 函数,我们可以探索和构造很多种滤波器结构。

FIR 滤波器结构

一个具有有限持续时间脉冲响应的滤波器的系统函数具有如下形式:

$$H(z)=b_0+b_1z^{-1}+\cdots+b_{M-1}z^{1-M}=\sum_{n=0}^{M-1}b_nz^{-n} \tag{6.5}$$

则脉冲响应 $h(n)$为:

$$h(n)=\begin{cases}b_n, & 0\leqslant n\leqslant M-1\\ 0, & \text{其它}\end{cases} \tag{6.6}$$

差分方程描述为:

$$y(n)=b_0x(n)+b_1x(n-1)+\cdots+b_{M-1}x(n-M+1) \tag{6.7}$$

是有限基底的线性卷积。

滤波器的阶数为 $M-1$,它的长度(等于系数的个数)为 M。FIR 滤波器总是稳定的,同 IIR 结构相比较而言,它们相对简单一些。而且,FIR 滤波器可设计成具有线性相位,这是某些应用所希望的。

我们将考虑下面四种结构:

- **直接形式**:这种形式以给定的形式直接实现差分方程(6.7)。

• **级联形式**:这种形式把(6.5)中的系统函数 $H(z)$因式分解成二阶因子,然后用级联连接实现。

• **线性相位形式**:当 FIR 滤波器具有线性相位响应时,它的脉冲响应出现一定的对称性。我们将在这种形式下研究对称关系,把乘法的计算量缩至一半。

• **频率采样形式**:这种结构基于脉冲响应 $h(n)$的 DFT,这会导致一个并联结构。采用基于频率响应 $H(e^{j\omega})$采样的设计技术也是可行的。

我们将用一些例子简要叙述上述四种形式。上节开发的 MATLAB 函数 dir2cas 对这里的级联形式也是适用的。

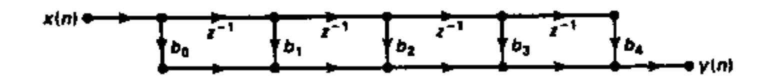


图 6.10 FIR 结构的直接形式

直接形式

由于在差分方程(6.7)中没有反馈回路,因此它由抽头延迟线实现。设 $M=5$(即,4 阶 FIR 滤波器),则:

$$y(n)=b_0x(n)+b_1x(n-1)+b_2x(n-2)+b_3x(n-3)+b_4x(n-4)$$

直接形式结构如图 6.10 所示。注意由于分母为 1,因此只有一种直接形式结构。

MATLAB 实现

在 MATLAB 中,FIR 结构的直接形式由包含系数$\{b_n\}$的行向量 b 描述。这种结构用 filter 函数实现,其中,把矢量 a 置为 1,这曾在第二章种讨论过。

级联形式

这种形式与 IIR 形式类似。把系统函数 $H(z)$转换成具有实系数的二阶子系统的乘积,子系统以直接形式实现,而整个滤波器用二阶子系统的级联实现。从(6.5)得:

$$\begin{aligned}H(z)&=b_0+b_1z^{-1}+\cdots+b_{M-1}z^{-M+1} \qquad (6.8)\\&=b_0\left(1+\frac{b_1}{b_0}z^{-1}+\cdots+\frac{b_{M-1}}{b_0}z^{-M+1}\right)\\&=b_0\prod_{k=1}^{K}\left(1+B_{k,1}z^{-1}+B_{k,2}z^{-2}\right)\end{aligned}$$

其中 K 等于$\left\lfloor\frac{M}{2}\right\rfloor$,实数 $B_{k,1}$、$B_{k,2}$表示二阶子系统的系数。$M=7$ 时,结构形式如图 6.1 所示。

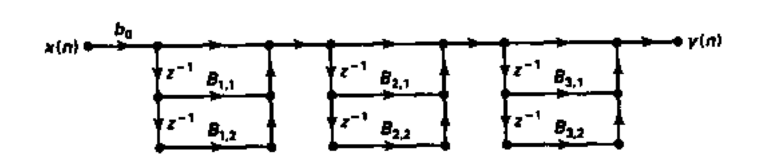


图 6.11 FIR 结构的级联形式

MATLAB 实现

尽管可以为 FIR 级联形式开发一个新的 MATLAB 函数，但我们还是使用 dir2cas 函数，把分母矢量 a 置为 1。类似地，用 cas2dir 函数可从级联形式得到直接形式。

线性相位形式

对选频滤波器(例如，低通滤波器)，通常希望得到线性相位，它为频率的线性函数；也就是，希望得到

$$\angle H(e^{j\omega}) = \beta - \alpha\omega, \quad -\pi < \omega \leqslant \pi \tag{6.9}$$

其中 $\beta = 0$ 或 $\pm\pi/2$，α 为一个常数。对于因果 FIR 滤波器，它的脉冲响应在区间 $[0, M-1]$ 上，线性相位条件(6.9)揭示了 $h(n)$ 有下面的对称性：

$$h(n) = h(M-1-n);\ \beta = 0, 0 \leqslant n \leqslant M-1 \tag{6.10}$$

$$h(n) = -h(M-1-n);\ \beta = \pm\pi/2, 0 \leqslant n \leqslant M-1 \tag{6.11}$$

满足条件(6.10)的脉冲响应叫做对称脉冲响应，而满足(6.11)的叫做反对称脉冲响应。这些对称条件可在称为线性相位的结构中使用。

若差分方程(6.7)具有式(6.10)中对称脉冲响应，即

$$\begin{aligned} y(n) &= b_0 x(n) + b_1 x(n-1) + \cdots + b_1 x(n-M+2) + b_0 x(n-M+1) \\ &= b_0[x(n) + x(n-M+1)] + b_1[x(n-1) + x(n-M+2)] + \cdots \end{aligned}$$

图 6.12 表示了 M 为奇数和偶数两种情况下实现上述差分方程的方框图。

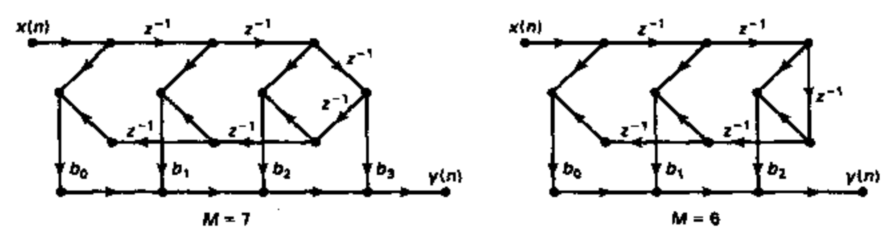


图 6.12 FIR 结构的线性相位形式(对称脉冲响应)

显然，这种结构比直接形式所需的乘法数小 50%。对反对称脉冲响应可导出类似的结构。

MATLAB 实现

线性相位结构在本质上为直接形式，它只是缩减了乘法计算量。因此，在 MATLAB 实现上，线性相位结构等于直接形式。

例 6.4 FIR 滤波器由下列系统函数给定

$$H(z)=1+16\frac{1}{16}z^{-4}+z^{-8}$$

求出并画出直接形式、线性相位形式和级联形式结构。

a. 直接形式： 差分方程为：

$$y(n)=x(n)+16.0625x(n-4)+x(n-8)$$

直接形式结构如图 6.13(a)所示。

b. 线性相位形式： 差分方程可写为：

$$y(n)=[x(n)+x(n-8)]+16.0625x(n-4)$$

所得结构如图 6.13(b)所示。

c. 级联形式：

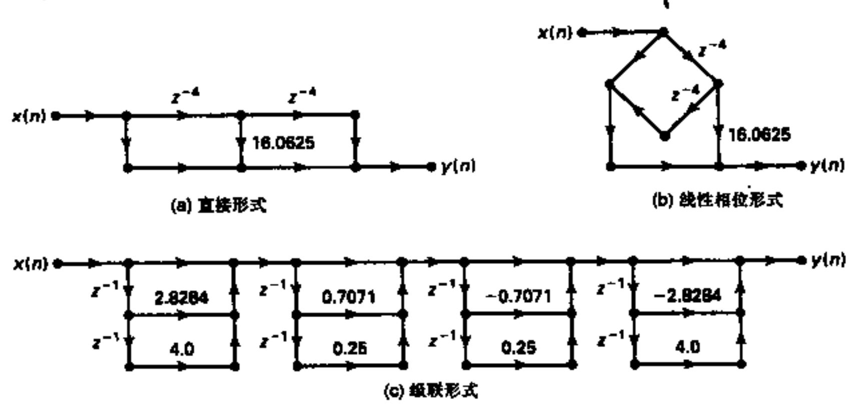


图 6.13 例 6.4 中的 FIR 滤波器结构

```
>> b=[1,0,0,0,16+1/16,0,0,0,1];
>> [b0,B,A]=dir2cas(b,1)
b0=1
B =
     1.0000     2.8284     4.0000
     1.0000     0.7071     0.2500
     1.0000    -0.7071     0.2500
```

```
    1.0000   -2.8284    4.0000
A =
    1   0   0
    1   0   0
    1   0   0
    1   0   0
```

级联形式结构如图 6.13(c)所示。

例 6.5 对例 6.4 中的滤波器,如果希望得到包含实系数线性相位分量的级联形式,它的结构该怎样?

解:我们感兴趣的是具有对称性和实系数的级联子系统。由线性相位 FIR 滤波器的性质(见第七章)可知,如果这样的滤波器在 $z=r\angle\theta$ 有一任意零点。为了得到实系数,则在 $(1/r)\angle\theta$、$r\angle-\theta$和$(1/r)\angle-\theta$必定存在三个其他的零点。现在我们利用这个性质,首先求出给定 8 阶多项式的零点,然后分出 4 个满足上述性质的零点,得到一个(4 阶)的线性相位子系统。可分出两个这样的子系统,把它们级联连接起来。

```
>> b=[1,0,0,0,16+1/16,0,0,0,1];
>> broots=roots(b)
broots =
  -1.4142 + 1.4142i
  -1.4142 - 1.4142i
   1.4142 + 1.4142i
   1.4142 - 1.4142i
  -0.3536 + 0.3536i
  -0.3536 - 0.3536i
   0.3536 + 0.3536i
   0.3536 - 0.3536i
>> B1=real(poly([broots(1),broots(2),broots(5),broots(6)]))
B1 =
    1.0000    3.5355    6.2500    3.5355    1.0000
>> B2=real(poly([broots(3),broots(4),broots(7),broots(8)]))
B2 =
    1.0000   -3.5355    6.2500   -3.5355    1.0000
```

这种结构如图 6.14 所示。

频率采样形式

在这种形式中,我们利用这样一个事实,即 FIR 滤波器的系统函数 $H(z)$可由它在单位圆

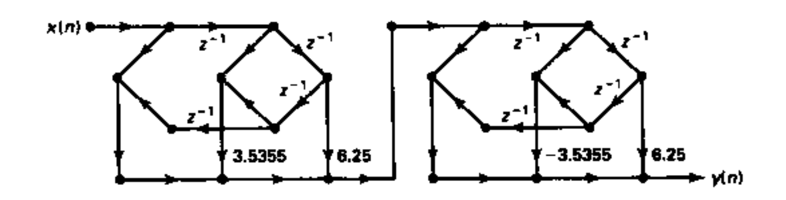


图 6.14　FIR 线性相位的级联结构

上的样本进行重构。从第五章对 DFT 的讨论可知，这些样本实际上是 M 点脉冲响应 $h(n)$ 的 M 点 DFT 值 $\{H(k), 0 \leqslant k \leqslant M-1\}$。因此有：

$$\begin{aligned} H(z) &= Z[h(n)] \\ &= Z[\mathrm{IDFT}\{H(k)\}] \end{aligned}$$

采用这种方法，得到[见(5.17)]：

$$H(z) = \left(\frac{1-z^{-M}}{M}\right) \sum_{k=0}^{M-1} \frac{H(k)}{1-W_M^{-k} z^{-1}} \tag{6.12}$$

由此说明在这种结构中，使用的是 DFT $H(k)$，而不是脉冲响应 $h(n)$（或者差分方程）。同时也可注意到一个有趣的现象，即，因为(6.12)包含零点和极点，由它描述的 FIR 滤波器具有与 IIR 滤波器相类似的递归形式。在 W_M^{-k} 处的极点与

$$1 - z^{-M} = 0$$

处的零点相抵销，因此所得的滤波器为 FIR 滤波器。(6.12)中的系统函数对应于一个并联结构，图 6.15 给出 $M=4$ 时的结构。

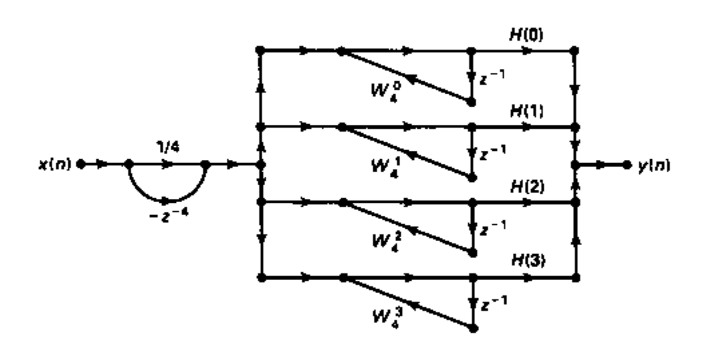


图 6.15　$M=4$ 的频率采样结构

图 6.15 的结构中存在的一个问题是它需要采用复数算法实现。由于 FIR 滤波器几乎总是实系数的，有可能得到仅采用实数算法的另一种实现。利用 DFT 的对称性和 W_M^{-k} 因子，可把(6.12)表示成(见习题 6.10)。

$$H(z)=\frac{1-z^{-M}}{M}\left\{\sum_{k=1}^{L}|H(k)2|H_k(z)+\frac{H(0)}{1-z^{-1}}+\frac{H(M/2)}{1+z^{-1}}\right\} \tag{6.13}$$

其中 M 为奇数时，$L=\frac{M-1}{2}$；M 为偶数时，$L=\frac{M}{2}-1$；$\{H_k(z),k=1,\ldots,L\}$是二阶子系统，由下式给出：

$$H_k(z)=\frac{\cos[\angle H(k)]-z^{-1}\cos[\angle H(k)-\frac{2\pi k}{M}]}{1-2z^{-1}\cos\left(\frac{2\pi k}{M}\right)+z^{-2}} \tag{6.14}$$

注意 DFT 样本 $H(0)$和 $H(M/2)$为实数，而且当 M 为奇数时，(6.13)中的第三项不存在。根据(6.13)和(6.14)，$M=4$ 时含实系数的频率采样结构如图 6.16 所示。

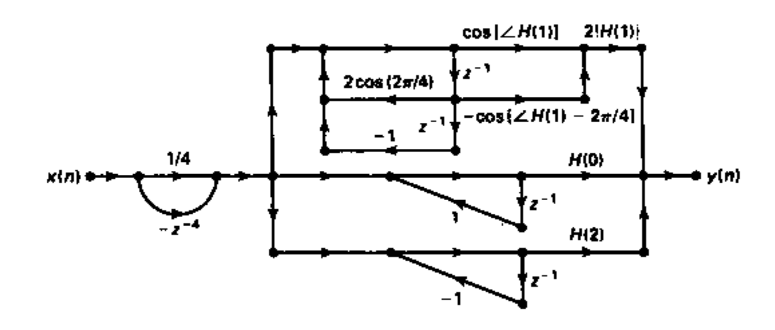


图 6.16 $M=4$ 时含实系数的频率采样结构

MATLAB 实现

给定脉冲响应 h(n)或 DFT $H(k)$，必须求出(6.13)和(6.14)中的系数。下面的 MATLAB 函数 dir2fs，通过直接实现(6.13)和(6.14)，把直接形式($h(n)$值)转换成频率采样形式。

```
function [C,B,A] = dir2fs(h)
% 直接型到频率采样型的转换
% ————————————————————
% [C,B,A] = dir2fs(h)
% C = 包含各并联部分增益的行向量
% B = 包含按行排列的分子系数矩阵
% A = 包含按行排列的分母系数矩阵
% h = FIR 滤波器的脉冲响应向量
%
M = length(h);
H = fft(h,M);
magH = abs(H); phaH = angle(H)';
```

```
% 检查 M 的奇偶性
if (M == 2 * floor(M/2))
    L = M/2 - 1; % M 为偶数
    A1 = [1, -1,0;1,1,0];
    C1 = [real(H(1)),real(H(L+2))];
else
    L = (M - 1)/2; % M 为奇数
    A1 = [1, -1,0];
    C1 = [real(H(1))];
end
k = [1:L]';
% 初始化 B 和 A 数组
B = zeros(L,2); A = ones(L,3);
% 计算分母系数
A(1:L,2) = -2 * cos(2 * pi * k/M); A = [A;A1];
% 计算分子系数
B(1:L,1) = cos(phaH(2:L+1));
B(1:L,2) = -cos(phaH(2:L+1) - (2 * pi * k/M));
% 计算增益系数
C = [2 * magH(2:L+1),C1]';
```

在上述函数中，脉冲响应值由数组 h 提供。经过变换以后，数组 C 中将包含每个并联子系统的增益值，首先给出二阶并联子系统的增益，随后为 $H(0)$和 $H(M/2)$(如果 M 为偶数)。矩阵 B 中为分子系数，每个二阶环节是长度为 2 的行向量。矩阵 A 为分母系数，每个二阶环节是长度为 3 的行向量，分别与 B 各行相对应，随后是一阶子系统的系数。

图 6.16 所示的结构在实际中会遇到一个问题，由于它在单位圆上存在极点。这将导致滤波器临界不稳定。如果滤波器不受激于其某个极点频率，则输出是有界的。为避免这个问题，对 $H(z)$在圆$|z|=r$ 上采样，其中半径 r 非常接近于 1，但比 1 小(如 $r=0.99$)，得到：

$$H(z)=\frac{1-r^{M}z^{-M}}{M}\sum_{k=0}^{M-1}\frac{H(k)}{1-rW_{M}^{-k}z^{-k}};\ H(k)=H(re^{j2\pi k/M}) \tag{6.15}$$

现在当 $r\approx 1$ 时，近似地使 $H(re^{j2\pi k/M})\approx H(e^{j2\pi k/M})$，我们得到一个与包含实数值的图 6.16 所示相似的稳定结构。这会在习题 6.11 中探讨。

例 6.6 设 $h(n)=\frac{1}{9}\{\underset{\uparrow}{1},2,3,2,1\}$，求出并画出频率采样结构。

解：MATLAB 程序

```
>> h = [1,2,3,2,1]/9;
>> [C,B,A] = dir2fs(h)
```

```
C =
    0.5818
    0.0849
    1.0000
B =
   -0.8090      0.8090
    0.3090     -0.3090
A =
    1.0000     -0.6180     1.0000
    1.0000      1.6180     1.0000
    1.0000     -1.0000          0
```

由于 $M=5$ 是奇数，因此只有一个一阶环节。从而

$$H(z)=\frac{1-z^{-5}}{5}\Big[0.5818\frac{-0.809+0.809z^{-1}}{1-0.618z^{-1}+z^{-2}} +0.0848\frac{0.309-0.309z^{-1}}{1+1.618z^{-1}+z^{-2}}+\frac{1}{1-z^{-1}}\Big]$$

频率采样形式如图 6.17 所示。

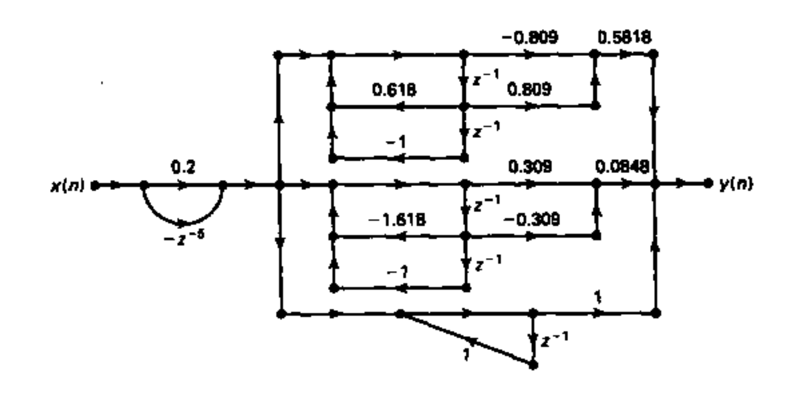


图 6.17 例 6.6 中的频率采样结构

例 6.7 32 点线性相位 FIR 滤波器的频率样本由下式给定：

$$|H(k)|=\begin{cases}1, & k=0,1,2\\ 0.5, & k=3\\ 0, & k=4,5,\ldots,15\end{cases}$$

求出它的频率采样形式，并比较它与线性相位形式的计算复杂度。

解：在本例中，由于给定 DFT $H(k)$ 的样本，因此可直接根据(6.13)和(6.14)确定结构。

但这里我们将使用 dir2fs 函数,故必须求出脉冲响应 $h(n)$。利用对称性和线性相位约束,把 DFT $H(k)$重新组合为:

$$H(k)=|H(k)|e^{j\angle H(k)},\ k=0,1,\ldots,31$$
$$|H(k)|=|H(32-k)|,\ k=1,2,\ldots,31;H(0)=1$$
$$\angle H(k)=-\frac{31}{2}\frac{2\pi}{32}k=-\angle H(32-k),\ k=0,1,\ldots,31$$

现在由 $H(k)$的 IDFT 可得到所希望的脉冲响应。

```
>> M = 32; alpha = (M - 1)/2;
>> magHk = [1,1,1,0.5,zeros(1,25),0.5,1,1];
>> k1 = 0:15; k2 = 16:M - 1;
>> angHk = [ - alpha * (2 * pi)/M * k1, alpha * (2 * pi)/M * (M - k2)];
>> H = magHk. * exp(j * angHk);
>> h = real(ifft(H,M));
>> [C,B,A] = dir2fs(h)
C =
    2.0000
    2.0000
    1.0000
    0.0000
    0.0000
    0.0000
    0.0000
         0
    0.0000
    0.0000
    0.0000
    0.0000
    0.0000
    0.0000
    0.0000
    1.0000
         0
B =
   -0.9952    0.9952
    0.9808   -0.9808
   -0.9569    0.9569
   -0.8944    0.3162
```

```
    0.9794   -0.7121
    0.8265    0.2038
   -0.6754    0.8551
    1.0000    0.0000
    0.6866   -0.5792
    0.5191    0.9883
   -0.4430    0.4993
   -0.8944   -0.3162
   -0.2766    0.3039
    0.9343    0.9996
   -0.9077   -0.8084
A =
    1.0000   -1.9616    1.0000
    1.0000   -1.8478    1.0000
    1.0000   -1.6629    1.0000
    1.0000   -1.4142    1.0000
    1.0000   -1.1111    1.0000
    1.0000   -0.7654    1.0000
    1.0000   -0.3902    1.0000
    1.0000    0.0000    1.0000
    1.0000    0.3902    1.0000
    1.0000    0.7654    1.0000
    1.0000    1.1111    1.0000
    1.0000    1.4142    1.0000
    1.0000    1.6629    1.0000
    1.0000    1.8478    1.0000
    1.0000    1.9616    1.0000
    1.0000   -1.0000         0
    1.0000    1.0000         0
```

注意只有 4 个增益系数非零。因此频率采样形式为：

$$H(z)=\frac{1-z^{32}}{32}\left[2\,\frac{-0.9952+0.9952z^{-1}}{1-1.9616z^{-1}+z^{-2}}+2\,\frac{0.9808-0.9808z^{-1}}{1-1.8478z^{-1}+z^{-2}}+\frac{-0.9569+0.9569z^{-1}}{1-1.6629z^{-1}+z^{-2}}+\frac{1}{1-z^{-1}}\right]$$

为了求出计算复杂度，注意到 $H(0)=1$，故一阶子系统不需要乘法，而三个二阶子系统中的每个子系统需要三次乘法，则每个输出样本需要 9 次乘法。总共需要 13 次加法。实现线性相位结构每个输出样本需要 16 次乘法和 31 次加法，因此，这个 FIR 滤波器的频率采样结构

比线性相位结构更高效。

格型滤波器结构

格型滤波器广泛应用于数字语音处理和自适应滤波器实现中。在这些应用中，它比其他FIR或IIR滤波器结构更优越，这是因为在语音分析和语音合成中，它可用较少数目的系数为较大数目的谐振峰实时建模。FIR滤波器的格型滤波器描述是全零点格型，而IIR滤波器的描述是格型梯形结构。

全零点格型滤波器

长度为M(或阶数为$M-1$)的FIR滤波器具有$M-1$级格型结构，如图6.18所示。每一级的输入输出通过顺序递归方程建立联系[19]：

$$\begin{aligned} f_m(n) &= f_{m-1}(n) + K_m g_{m-1}(n-1), \quad m=1,2,\ldots,M-1 \\ g_m(n) &= K_m f_{m-1}(n) + g_{m-1}(n-1), \quad m=1,2,\ldots,M-1 \end{aligned} \tag{6.16}$$

其中，参数K_m，$m=1,2,\ldots,M-1$，叫做反射系数，它们是格型滤波器的系数。如果$f_m(n)$和$g_m(n)$的初值均为滤波器输入$x(n)$的倍数(乘以K_0)，则$M-1$级格型滤波器的输出与$(M-1)$阶FIR滤波器输出一致，也就是：

$$\begin{aligned} f_0(n) &= g_0(n) = K_0 x(n) \\ y(n) &= f_{M-1}(n) \end{aligned} \tag{6.17}$$

如果FIR滤波器以直接形式给出

$$H(z) = \sum_{m=0}^{M-1} b_m z^{-m} = b_0\left(1 + \sum_{m=1}^{M-1} \frac{b_m}{b_0} z^{-m}\right) \tag{6.18}$$

同时，把多项式$A_{M-1}(z)$表示为

$$A_{M-1}(z) = \left(1 + \sum_{m=1}^{M-1} a_{M-1}(m) z^{-m};\right) \tag{6.19}$$

$$a_{M-1}(m) = \frac{b_m}{b_0},\ m=1,\ldots,M-1$$

那么格型滤波器系数$\{K_m\}$可由下面的递归算法得到[19]：

$$
\begin{aligned}
K_0 &= b_0 \\
K_{M-1} &= \alpha_{M-1}(M-1) \\
J_m(z) &= z^{-m}A_m(z^{-1}); & m = M-1,\ldots,1 \\
A_{m-1}(z) &= \frac{A_m(z) - K_m J_m(z)}{1-K_m^2}, & m = M-1,\ldots,1 \\
K_m &= \alpha_m(m), & m = M-2,\ldots,1
\end{aligned}
\tag{6.20}
$$

注意如果对于任意 $m = 1,2,\ldots,M-1$，$|K_m| = 1$，上述算法失效。由于

$$
b_0 = |b_{M-1}| \Rightarrow |K_{M-1}| = |\alpha_{M-1}(M-1)| = \left|\frac{b_{M-1}}{b_0}\right| = 1
$$

可见线性相位滤波器满足此条件。因此它不能用格型滤波器实现。

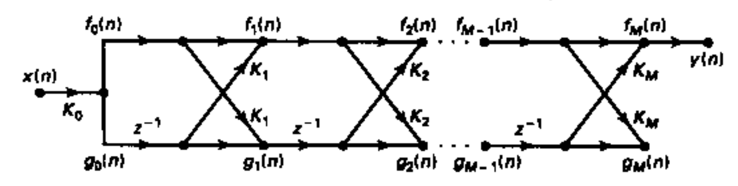


图 6.18　全零格型滤波器

MATLAB 实现

给定直接形式的系数 $\{b_n\}$，用(6.20)可得到格型滤波器系数 $\{K_m\}$。这可由下面的 MATLAB 函数 dir2latc 完成。注意计算 $J_m(z)$ 的方程暗示了多项式 $J_m(z)$ 是对多项式 $A_m(z)$ 进行 fliplr 运算而得到的。

```
function [K] = dir2latc(b)
% FIR 的直接形式到全零点格型的形式转换
% ————————————————————————
% [K] = dir2latc(b)
% K = 格型滤波器系数(反射系数)
% b = 直接形式的多项式系数(脉冲响应)
%
M = length(b);
K = zeros(1,M);
b1 = b(1);
if b1 == 0
error('b(1) 等于零')
end
K(1) = b1; A = b/b1;
```

```
for m = M: - 1:2
K(m) = A(m);
J = fliplr(A);
A = (A - K(m) * J)/(1 - K(m) * K(m));
A = A(1:m - 1);
end
```

格型滤波器可由(6.16)和(6.17)实现,具体由下面的 latcfilt 函数完成。

```
function [y] = latcfilt(K,x)
% 滤波器的格型实现
% ————————————
% y = latcfilt(K,x)
% y = 输出序列
% K = 格型滤波器(反射)系数数组
% x = 输入序列
%
Nx = length(x) - 1;
x = K(1) * x;
M = length(K) - 1; K = K(2:M + 1);
fg = [x; [0 x(1:Nx)]];
for m = 1:M
     fg = [1,K(m);K(m),1] * fg;
     fg(2,:) = [0 fg(2,1:Nx)];
end
y = fg(1,:);
```

利用递归过程[19],也可用方程(6.20)来从格型滤波器系数{ K_m }求出直接形式系数{ b_m }:

$$
\begin{aligned}
A_0(z) &= J_0(z) = 1 \\
A_m(z) &= A_{m-1}(z) + K_m z^{-1} J_{m-1}(z), \quad & m = 1,2,\ldots,M-1 \\
J_m(z) &= z^{-m} A_m(z^{-1}), \quad & m = 1,2,\ldots,M-1 \\
b_m &= K_0 a_{M-1}(m), \quad & m = 0,1,\ldots,M-1
\end{aligned}
\tag{6.21}
$$

下面给出的 MATLAB 函数 latc2dir 可实现(6.21),注意乘积 $K_m z^{-1} J_{m-1}(z)$ 通过两个相应数组的卷积得到,而多项式 $J_m(z)$ 由对多项式 $A_m(z)$ 进行 fliplr 运算得到。

```
function [b] = latc2dir(K)
```

```
% 全零点格型到 FIR 的直接形式的形式转换
% ————————————————————
% [b] = latc2dir(K)
% b = 直接形式的多项式系数(脉冲响应)
% K = 格型滤波器系数(反射系数)
%
M = length(K);
J = 1;  A = 1;
for m = 2:1:M
A = [A,0] + conv([0,K(m)],J);
J = fliplr(A);
end
b = A * K(1);
```

例 6.8　FIR 滤波器由下面的差分方程给定

$$y(n) = 2x(n) + \frac{13}{12}x(n-1) + \frac{5}{4}x(n-2) + \frac{2}{3}x(n-3)$$

求出它的格型形式。

解:MATLAB 程序——————————————————

```
>> b = [2,13/12,5/4,2/3];
>> K = dir2latc(b)
K =
    2.0000    0.2500    0.5000    0.3333
```

因此

$$K_0 = 2, K_1 = \frac{1}{4}, K_2 = \frac{1}{2}, K_3 = \frac{1}{3}$$

直接形式和格型形式如图 6.19 所示。为了检查格型结构是否正确,用两种形式计算滤波器的脉冲响应。

```
>> [x,n] = impseq(0,0,3);
>> format long
>> hdirect = filter(b,1,delta)
hdirect =
  2.00000000000000  1.08333333333333  1.25000000000000  0.66666666666667
```

```
>>hlattice = latcfilt(K,delta)
hlattice =
  2.00000000000000   1.08333333333333   1.25000000000000   0.66666666666667
```

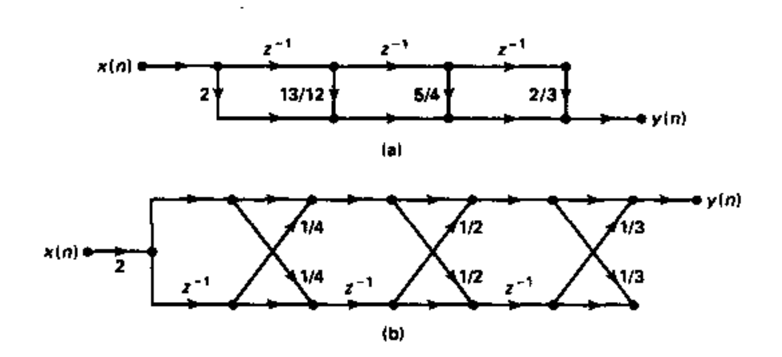


图 6.19　例 6.8 中的 FIR 滤波器结构(a)直接形式 (b)格型形式

全极点格型滤波器

IIR 滤波器的格型结构受限制于全极点的系统函数。可根据 FIR 格型结构来开发它。设一个全极点系统函数由下式给定：

$$H(z) = \frac{1}{1 + \sum_{m=1}^{M} a_N(m) z^{-m}} \tag{6.22}$$

与(6.19)相比，可知 $H(z) = \frac{1}{A_N(z)}$。显然，它是图 6.18 所示的 FIR 格型结构的逆系统(除因子 b_0 外)，这个 N 阶 IIR 滤波器具有如图 6.20 所示的 N 级格型结构。每一级的输入输出通过顺序递归方程[19]建立联系：

$$\begin{aligned} f_N(n) &= x(n) \\ f_{m-1}(n) &= f_m(n) - K_m g_{m-1}(n-1), \quad m = N, N-1, \ldots, 1 \\ g_m(n) &= K_m f_{m-1}(n) + g_{m-1}(n-1), \; m = N, N-1, \ldots, 1 \\ y(n) &= f_0(n) = g_0(n) \end{aligned} \tag{6.23}$$

其中参数 $K_m, m = 1, 2, \ldots, M-1$，是全极点格型的反射系数，除了 K_0 等于 1 外，其余各系数可通过(6.20)得到。

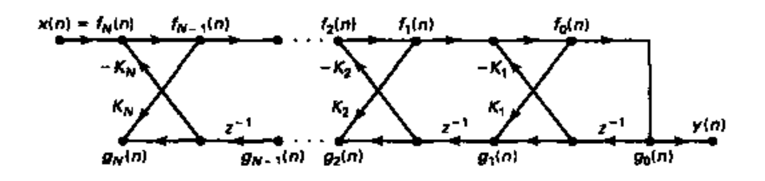


图 6.20　全极点格型滤波器

MATLAB 实现

既然,与 FIR 格型滤波器相同,IIR 滤波器系数同样可由(6.20)导出。因此我们可使用 MATLAB 中的 dir2latc 函数。必须记住忽略数组 K 中的系数 K_0。类似地,假定把数组 K 中的第一个元素设为 $K_0 = 1$,则可使用 latc2dir 函数把格型 $\{K_m\}$ 系数转换成直接形式 $\{a_N(m)\}$。IIR 格型的实现可用(6.23)式。我们将在下一节中讨论它。

例 6.9 全极点 IIR 滤波器由下式给出:

$$H(z)=\frac{1}{1+\frac{13}{24}z^{-1}+\frac{5}{8}z^{-2}+\frac{1}{3}z^{-3}}$$

求出它的格型结构。

解:MATLAB 程序

```
>> a=[1,13/24,5/8,1/3];
>> K=dir2latc(b)
K =
     1.0000     0.2500     0.5000     0.3333
```

因此

$$K_1=\frac{1}{4},\quad K_2=\frac{1}{2},\quad \text{and}\quad K_3=\frac{1}{3}$$

IIR 滤波器的格型形式和直接形式如图 6.21 所示。

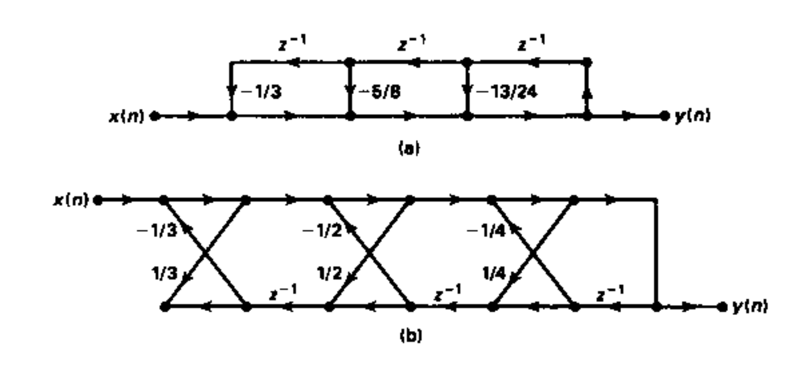


图 6.21 例 6.9 中的 IIR 滤波器结构:(a) 直接形式 (b)格型形式

格型梯形滤波器

一般的 IIR 滤波器既包含零点,又包括极点,它可用全极点格型作为基本构造模块,以格

型结构实现。假定 IIR 滤波器的系统函数为

$$H(z)=\frac{\sum_{k=0}^{M}b_M(k)z^{-k}}{1+\sum_{k=1}^{N}a_N(k)z^{-k}}=\frac{B_M(z)}{A_N(z)} \tag{6.24}$$

其中,不失一般性,假设 $N\geqslant M$。为构造一个格型类型结构,首先根据(6.24)的分母,以系数 $K_m,1\leqslant m\leqslant N$ 实现全极点格型,然后,增加一个梯形部分。把输出看作{ $g_m(n)$ }的加权线性组合。如图 6.22 所示是 $M=N$ 时的情况。其结果为具有格型梯形结构的零点-极点 IIR 滤波器。它的输出为:

$$y(n)=\sum_{m=0}^{M}C_m g_m(n) \tag{6.25}$$

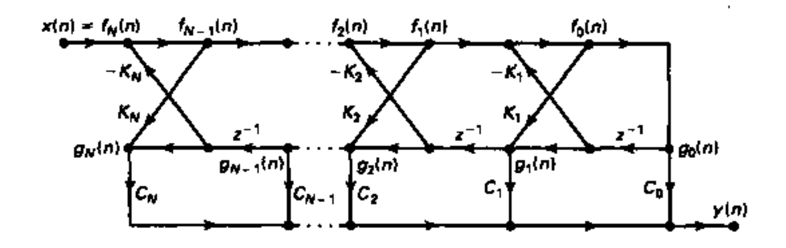


图 6.22　实现零点-极点 IIR 滤波器的格型梯形结构

其中{ C_m }称为确定系统函数 $H(z)$ 零点的梯形系数。参考文献[19]中已证明,{ C_m }由下式给定:

$$B_M(z)=\sum_{m=0}^{M}C_m J_m(z) \tag{6.26}$$

其中 $J_m(z)$ 是(6.20)中的多项式。由(6.26)可得到递归关系

$$B_m(z)=B_{m-1}(z)+C_m J_m(z);\quad m=0,2,\ldots,M$$

或相应地,由 $B_m(z)$ 和 $A_m(z)$ 的定义得到:

$$C_m=b_m+\sum_{i=m+1}^{M}C_i\alpha_i(i-m);\ m=M,M-1,\ldots,0 \tag{6.27}$$

MATLAB 实现

为实现一般有理 IIR 滤波器的格型梯形结构,首先用递归关系(6.20)从 $A_N(z)$ 得到格型

系数{ K_m },然后对(6.27)递归地求解得到梯形系数{ C_m },实现分子 $B_M(z)$。这由下面的 MATLAB 函数 dir2ladr 完成。当数组 b 置为 $b=[1]$时,它也可用来求出全极点格型参数。

```
function [K,C] = dir2ladr(b,a)
% 直接型到极点-零点格型/梯形的形式转换
% ——————————————————————————
% [K,C] = dir2ladr(b,a)
% K = 格型系数(反射系数), [K1,...,KN]
% C = 梯形系数, [C0,...,CN]
% b = 直接型的分子多项式系数
% a = 直接型的分母多项式系数
%
a1 = a(1); a = a/a1; b = b/a1;
M = length(b); N = length(a);
if M > N
    error(' * * * b的长度必须 < = a 的长度 * * *')
end
b = [b, zeros(1,N - M)]; K = zeros(1,N - 1);
A = zeros(N - 1,N - 1); C = b;
for m = N - 1: - 1:1
    A(m,1:m) = - a(2:m + 1) * C(m + 1);
    K(m) = a(m + 1);
    J = fliplr(a);
    a = (a - K(m) * J)/(1 - K(m) * K(m));
    a = a(1:m);
    C(m) = b(m) + sum(diag(A(m:N - 1,1:N - m)));
end
```

注意:为了使用此函数,必须 $N \geqslant M$。如果 $M > N$,则分子 $B_m(z)$应除以分母 $A_N(z)$,得到纯有理部分和多项式部分。纯有理部分可用格型梯形结构实现,同时,多项式部分由直接结构实现。

为了把格型梯形形式转换成直接形式,首先对系数{ K_m }用(6.21)中的递归过程来求出{ $a_N(k)$ },然后递归地求解(6.27)得到{ $b_M(k)$ }。这可由下面的 MATLAB 函数 ladr2dir 完成。

```
function [b,a] = ladr2dir(K,C)
% 格型/梯形到直接型的形式转换
% [b,a] = ladr2dir(K,C)
% ——————————————
% b = 直接型的分子多项式系数
```

```
% a=直接型的分母多项式系数
% K=格型系数(反射系数),
% C=梯形系数,
%
N=length(K); M=length(C);
C=[C, zeros(1,N-M+1)];
J=1; a=1; A=zeros(N,N);
for m=1:1:N
    a=[a,0]+conv([0,K(m)],J);
    A(m,1:m)=-a(2:m+1);
    J=fliplr(a);
end
b(N+1)=C(N+1);
for m=N:-1:1
    A(m,1:m)=A(m,1:m)*C(m+1);
    b(m)=C(m) - sum(diag(A(m:N,1:N-m+1)));
end
```

用(6.23)和(6.25)可实现格型梯形滤波器。这由下面给出的 MATLAB 函数 ladrfilt 完成。应该注意到。由于这种实现的递归特性和反馈环的影响,此 MATLAB 函数既不高明又欠效率。在实现这种格型梯形结构时,不可能利用 MATLAB 内在的并行处理能力。

```
function [y]=ladrfilt(K,C,x)
% IIR 滤波器的格型梯形结构实现
% ----------------------------
% [y]=ladrfilt(K,C,x)
% y=输出序列
% K=LATTICE (反射)系数数组
% C=LADDER 系数数组
% x=输入序列
%
Nx=length(x); y=zeros(1,Nx);
N=length(C); f=zeros(N,Nx); g=zeros(N,Nx+1);
f(N,:)=x;
for n=2:1:Nx+1
    for m=N:-1:2
        f(m-1,n-1)=f(m,n-1) - K(m-1)*g(m-1,n-1);
        g(m,n)=K(m-1)*f(m-1,n-1) + g(m-1,n-1);
  end
  g(1,n)=f(1,n-1);
```

```
end
y = C * g(:,2:Nx+1);
```

例 6.10 把下面的零点-极点 IIR 滤波器转换成格型梯形结构。

$$H(z)=\frac{1+2z^{-1}+2z^{-2}+z^{-3}}{1+\frac{13}{24}z^{-1}+\frac{5}{8}z^{-2}+\frac{1}{3}z^{-3}}$$

解: MATLAB 程序

```
>> b=[1,2,2,1]; a=[1,13/24,5/8,1/3];
>> [K,C] = dir2ladr(b,a)
K =
    0.2500    0.5000    0.3333
C =
   -0.2695    0.8281    1.4583    1.0000
```

因此

$$K_1=\frac{1}{4},K_2=\frac{1}{2},K_3=\frac{1}{3}$$

并且

$$C_0=-0.2695,C_1=0.8281,C_2=1.4583,C_3=1$$

所得的直接形式和格型梯形形式结构如图 6.23 所示。为检查格型梯形结构是否正确,让我们用两种形式计算它的脉冲响应的前 8 个样本。

```
>> [x,n] = impseq(0,0,7);format long
>> hdirect = filter(b,a,x)
hdirect =
  Columns 1 through 4
  1.00000000000000   1.45833333333333   0.58506944444444   -0.56170428240741
  Columns 5 through 8
  -0.54752302758488   0.45261700163162   0.28426911049255   -0.25435705167494
>> hladder = ladrfilt(K,C,x)
hladder =
  Columns 1 through 4
  1.00000000000000   1.45833333333333   0.58506944444444   -0.56170428240741
```

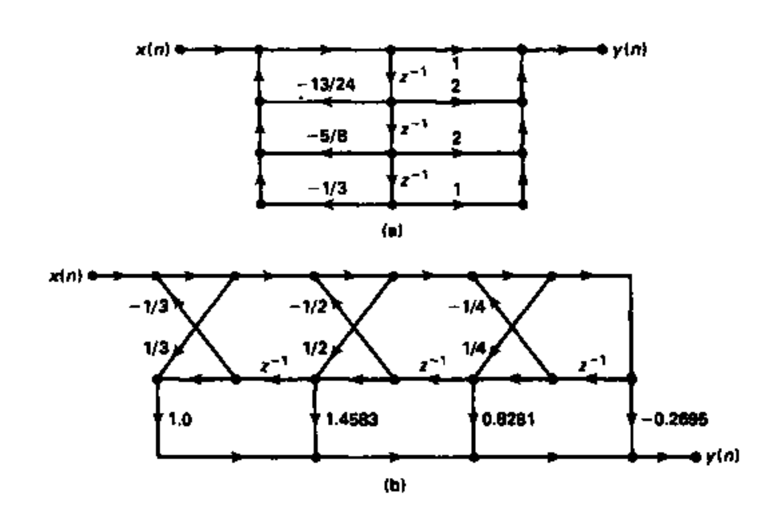


图 6.23 例 6.10 中的 IIR 滤波器结构：
(a)直接形式 (b)格型梯形形式

Columns 5 through 8

−0.54752302758488 0.45261700163162 0.28426911049255 −0.25435705167494

习　题

P6.1 一个因果、线性、时不变系统为

$$y(n)=\sum_{k=0}^{5}\left(\frac{1}{2}\right)^{k}x(n-k)+\sum_{l=1}^{5}\left(\frac{1}{3}\right)^{l}y(n-l)$$

确定并画出下列结构的方框图。用下面每种情况下的结构，计算系统对

$$x(n)=u(n),0\leqslant n\leqslant 100$$

的响应。

a．直接Ⅰ型

b．直接Ⅱ型

c．包含二阶直接Ⅱ型子系统的级联形式

d．包含二阶直接Ⅱ型子系统的并联形式

e．格型梯形形式

P6.2 一个 IIR 滤波器由下面的系统函数描述：

$$H=2\left(\frac{1+0z^{-1}+z^{-2}}{1-0.8z^{-1}+0.64z^{-2}}\right)\left(\frac{2-z^{-1}}{1-0.75z^{-1}}\right)\left(\frac{1+2z^{-1}+z^{-2}}{1+0.81z^{-2}}\right)$$

确定并画出下列结构。

a．直接Ⅰ型

b．直接Ⅱ型

c．包含二阶直接Ⅱ型子系统的级联形式

d．包含二阶直接Ⅱ型子系统的并联形式

e．格型梯形形式

P6.3 一个IIR滤波器由下面的系统函数描述：

$$H(z)=\left(\frac{-14.75-12.9z^{-1}}{1-\frac{7}{8}z^{-1}+\frac{3}{32}z^{-2}}\right)+\left(\frac{24.5+26.82z^{-1}}{1-z^{-1}+\frac{1}{2}z^{-2}}\right)$$

确定并画出下列结构。

a．直接Ⅰ型

b．直接Ⅱ型

c．包含二阶直接Ⅱ型子系统的级联形式

d．包含二阶直接Ⅱ型子系统的并联形式

e．格型梯形形式

P6.4 图6.24给出一个因果线性时不变系统，确定并画出下列结构：

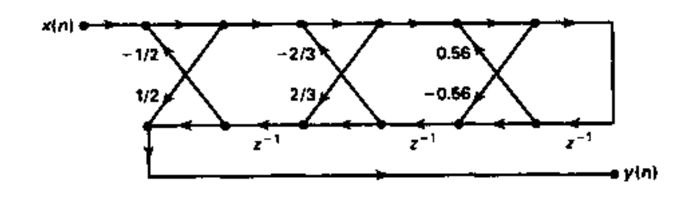


图6.24 习题6.4中的结构

a．直接Ⅰ型

b．直接Ⅱ型

c．包含二阶直接Ⅱ型子系统的级联形式

d．包含二阶直接Ⅱ型子系统的并联形式

P6.5 一个线性时不变系统具有系统函数

$$H(z)=\frac{0.5(1+z^{-1})^{6}}{\left(1-\frac{3}{2}z^{-1}+\frac{7}{8}z^{-2}-\frac{13}{16}z^{-3}-\frac{1}{8}z^{-4}-\frac{11}{32}z^{-5}+\frac{7}{16}z^{-6}\right)}$$

它由如图6.25所示的流图实现。

a．填出方框图中的系数。

b．你的解唯一吗？请说明。

P6.6 一个线性时不变系统具有系统函数

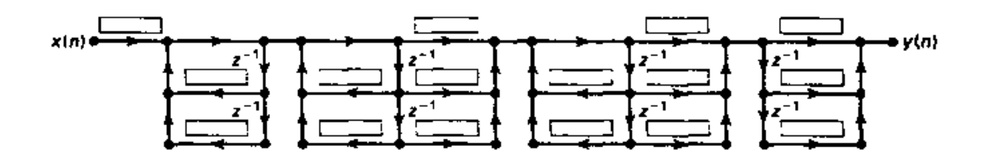


图 6.25 习题 6.5 中的结构

$$H(z)=\frac{5+11.2z^{-1}+5.44z^{-2}-0.384z^{-3}-2.3552z^{-4}-1.2288z^{-5}}{1+0.8z^{-1}-0.512z^{-3}-0.4096z^{-4}}$$

它由如图 6.26 所示的流图实现。填出方框图中的系数。

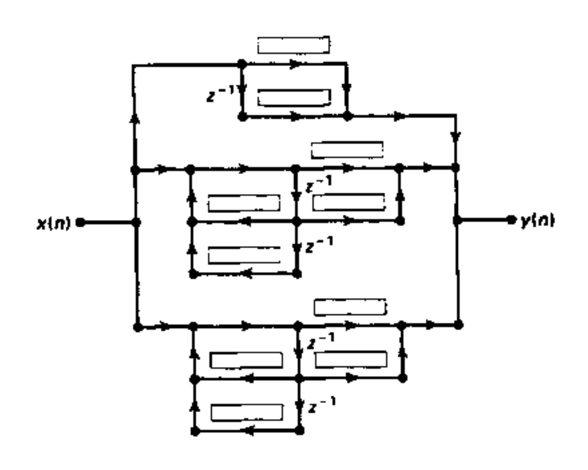


图 6.26 习题 6.6 中的结构

P6.7 考察习题 6.5 中的线性时不变系统

$$H(z)=\frac{0.5(1+z^{-1})^{6}}{\left(1-\frac{3}{2}z^{-1}+\frac{7}{8}z^{-2}-\frac{13}{16}z^{-3}-\frac{1}{8}z^{-4}-\frac{11}{32}z^{-5}+\frac{7}{16}z^{-6}\right)}$$

它由如图 6.27 所示的流图实现。

a．填出方框图中的系数。

b．你的解唯一吗？请说明

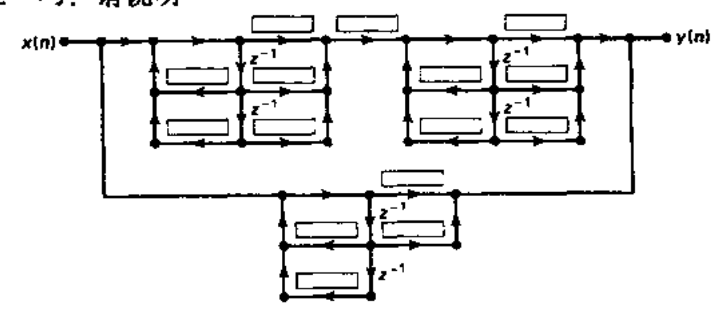


图 6.27 习题 6.7 中的结构

P6.8 FIR 滤波器由下面的差分方程描述：

$$y(n)=\sum_{k=0}^{10}\left(\frac{1}{2}\right)^{|5-k|}x(n-k)$$

确定并画出下列结构的方框图。

a．直接形式

b．线性相位形式

c．级联形式

d．频率采样形式

P6.9 线性时不变系统由下面的系统函数给定：

$$H(z)=\sum_{k=0}^{10}(2z)^{-k}$$

确定并画出下列结构的方框图。

a．直接形式

b．级联形式

c．格型形式

d．频率采样形式

P6.10 利用 DFT 的共轭对称性

$$H(k)=\begin{cases}H(0), & k=0\\ H^*(M-k), & k=1,\dots,M-1\end{cases}$$

和因子 W_M^{-k} 的共轭对称性，证明对实数滤波器(6.12)可写成(6.13)和(6.14)的形式。

P6.11 在频率采样结构中，为了避免单位圆上的极点，正如在本章中讨论的，对 $H(z)$在 $z_k=re^{j2\pi k/M}, k=0,\dots,M-1$，上采样，其中 $r\approx 1$(但<1)。

a．利用

$$H(re^{j2\pi k/M})\approx H(k)$$

证明频率采样结构由下式给出：

$$H(z)=\frac{1-(rz)^{-M}}{M}\left\{\sum_{k=1}^{L}2\,|H(k)|\,H_k(z)+\frac{H(0)}{1-rz^{-1}}+\frac{H(M/2)}{1+rz^{-1}}\right\}$$

其中

$$H_k(z)=\frac{\cos[\angle H(k)]-rz^{-1}\cos[\angle H(k)-\frac{2\pi k}{M}]}{1-2rz^{-1}\cos\left(\frac{2\pi k}{M}\right)+z^{-2}},k=1,\ldots,L$$

M 为偶数。

b .修正 MATLAB 函数 dir2fs(在本章中开发的),以实现上述的频率采样结构,该函数的格式应该为:

```
function[C,B,A,rM] = dir2fs(h,r)
% 直接形式到频率采样形式的转换
% ————————————————————
% [C,B,A] = dir2fs(h)
% C = 包含各并联部分增益的行向量
% B = 包含按行排列的分子系数矩阵
% A = 包含按行排列的分母系数矩阵
% rM = 在前向反馈环中所需要的 r^M 因子
% h = FIR 滤波器的脉冲响应向量
% r = 进行采样的圆的直径(r < 1)
```

c .利用上述函数,求出例题 6.6 中脉冲响应的频率采样结构。

P6.12 一个 FIR 滤波器的格型参数为:

$$K_0=2,K_1=0.6,K_2=0.3,K_3=0.5,K_4=0.9$$

求出它的脉冲响应,并画出上述滤波器的直接形式和格型形式结构。

第七章 FIR 滤波器设计

现在我们把注意力转向按给定的指标设计系统的逆问题,这是很重要且很困难的一个问题。在数字信号处理中,有两种类型的重要系统。第一种类型的系统在时域中进行信号滤波,因此叫做数字滤波器。第二种系统提供频域里的信号表示,叫做频谱分析仪。在第五章中讲述了利用 DFT 的信号表示。在本章和下一章中,我们将学习 FIR 和 IIR 滤波器的几种基本设计算法。这些设计大多数是针对选频型的,也就是说,主要设计多通带低通、高通、带通和带阻滤波器。在 FIR 滤波器设计中,还会涉及到像微分器或希尔伯特变换器之类的系统,尽管它们不是选频滤波器,但是也遵循选频滤波器的设计技术。更完善的滤波器设计则基于任意频域指标,研究它所需要的工具超出了本书的范围。

首先,我们给出一些关于设计原理和指标的基本知识。这些知识同时适用于 FIR 和 IIR 滤波器设计。然后,将研究 FIR 滤波器的设计算法。在第八章中,对 IIR 滤波器将作类似的研究。

基本知识

数字滤波器的设计要经过三步完成:

- **确定指标**:在设计一个滤波器前,必须有一些指标。这些指标要根据应用确定。
- **逼近**:一旦确定了技术指标,就可利用学习过的数学和基本原理,提出一个滤波器模型来逼近给定的一套指标。这一步是滤波器设计所要研究的问题。
- **实现**:上面的两步的结果是得到以差分方程,或系统函数,或脉冲响应 $h(n)$ 描述的滤波器。根据这个描述,可以用硬件或如第六章所讨论的在计算机上用软件实现它。在本章及下一章中,我们将详细讨论第二步,即把指标转换成一个滤波器描述。

在很多实际应用,像语音或音频信号处理中,数字滤波器用来实现选频操作。因此,指标的形式应为频域中的幅度和相位响应。在通带中,通常期望具有线性相位响应。正如在第六章中讨论的,在 FIR 滤波器中,可以得到精确的线性相位。在 IIR 滤波器中,通带中的线性相位是不可能得到的,因此,只考虑幅度指标。

幅度指标可以两种方式给出。第一种,叫做绝对指标,它提供对幅度响应函数 $|H(e^{j\omega})|$ 的要求。这些指标一般用于 FIR 滤波器。IIR 滤波器则以一种不同的方式给出指标,我们将在第八章中讨论它。第二种方法叫做相对指标,它以分贝(dB)值的形式提出要求,其值定义为:

$$\text{dB 值} = -20\log_{10}\frac{|H(e^{j\omega})|}{|H(e^{j\omega})|_{\max}} \geqslant 0$$

在实际中,这种方法最受欢迎。为了说明这些指标,我们以低通滤波器设计为例进行讨论。

绝对指标

低通滤波器的典型绝对指标如图 7.1a 所示。其中

- $[0, \omega_p]$段叫做通带，δ_1 是在理想通带响应中能接受的容限(或波动)。
- $[\omega_s, \pi]$段叫做阻带，δ_2 是相应的容限。
- $[\omega_p, \omega_s]$叫做过渡带，在此段上对幅度响应没有限制。

相对指标(dB)

低通滤波器的典型相对技术指标如图 7.1b 所示。其中

- R_p 是通带波动，
- A_s 是阻带衰减的 dB 数

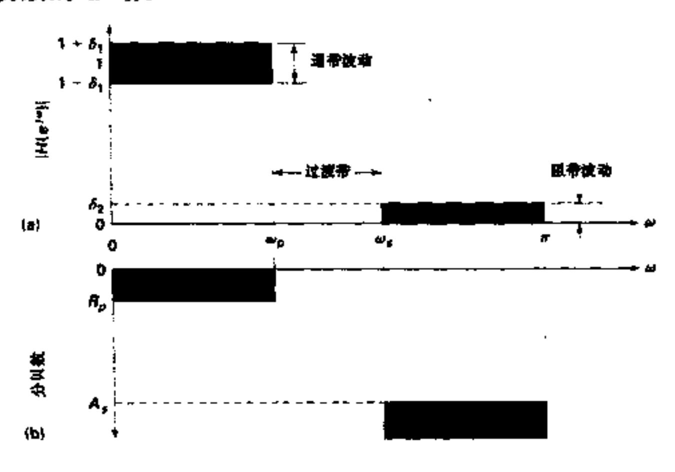


图 7.1　FIR 滤波器技术指标:(a)绝对 (b)相对

上述两种指标里给定的参数显然存在一定的关系。由于绝对指标中的$|H(e^{j\omega})|_{max}$等于$(1+\delta_1)$，因此，有

$$R_p = -20\log_{10}\frac{1-\delta_1}{1+\delta_1} > 0(\approx 0) \tag{7.1}$$

及

$$A_s = -20\log_{10}\frac{\delta_2}{1+\delta_1} > 0(\gg 1) \tag{7.2}$$

例 7.1　在一个滤波器技术指标中，通带波动为 0.25dB，阻带衰减为 50dB。确定 δ_1 和 δ_2。

解：利用(7.1)，得

$$R_p = 0.25 = -20\log_{10}\frac{1-\delta_1}{1+\delta_1} \Rightarrow \delta = 0.0144$$

利用(7.2),得

$$A_s = 50 = -20\log_{10}\frac{\delta_2}{1+\delta_1} = -20\log_{10}\frac{\delta_2}{1+0.0144} \Rightarrow \delta_2 = 0.0032$$

例 7.2 给定通带容限 $\delta_1 = 0.01$,阻带容限 $\delta_2 = 0.001$,确定通带波动和阻带衰减。

解:从(7.1)知,通带波动为:

$$R_p = -20\log_{10}\frac{1-\delta_1}{1+\delta_1} = 0.1737\text{dB}$$

从(7.1)知,通带波动为:

$$A_s = -20\log_{10}\frac{\delta_2}{1+\delta_1} = 60\text{dB}$$

上述指标是针对低通滤波器的,其他类型的选频滤波器,像高通和带通滤波器,它们的技术指标可作类似的给定。但是,最重要的设计参数是频带容限(或波动)和边缘频率。而给定的带为通带或阻带,相对而言是次要的。因此,在讨论设计技术时,我们将集中讨论低通滤波器,在下一章中讨论怎样把一个低通滤波器转换成其他类型的选频滤波器。因此开发并比较低通滤波器的设计技术是明智的。但是,我们也会提供其他类型滤波器的例子。鉴于以上讨论,总结出下面的设计目标。

问题描述 设计一个低通滤波器(即,得到它的系统函数 H(z)或差分方程),它的通带为 $[0, \omega_p]$,容限是 δ_1(或 R_p),阻带为 $[\omega_s, \pi]$,容限为 δ_2(或 A_s)。

在本章中,我们把注意力放在 FIR 数字滤波器的设计和逼近上。这些滤波器在设计和实现上具有一些优越性:

- 相位响应可为严格的线性。
- 由于不存在稳定性问题,所以设计相对简单。
- 实现极为经济。
- 在实现中,可使用 DFT。

正如在第六章中讨论的,一般来讲,我们感兴趣的是线性相位选频 FIR 滤波器。其优越性为:

- 在设计中,只包含实数算法,不涉及复数算法。
- 线性相位滤波器中,不存在延迟失真,只有固定数量的延迟。
- 长度为 M 的滤波器(阶数为 $M-1$),它的计算量为 $M/2$ 数量级。

我们首先讨论设计算法中所需要的线性相位 FIR 滤波器的性质。然后研究三种设计算法,分别叫做窗函数设计,频率采样设计和最优等波动设计技术。

线性相位 FIR 滤波器的性质

在这一节中,讨论线性相位滤波器的脉冲和频率响应的形状及其系统函数的零点位置,设

$h(n), 0 \leqslant n \leqslant M-1$ 是长度(或持续时间)为 M 的脉冲响应,那么,系统函数为:

$$H(z) = \sum_{n=0}^{M-1} h(n) z^{-n} = z^{-(M-1)} \sum_{n=0}^{M-1} h(n) z^{M-1-n}$$

它在原点 $z = 0$ 处有 $(M-1)$ 重极点和 $(M-1)$ 个位于 z 平面上任意位置的零点。它的频率响应函数为:

$$H(e^{j\omega}) = \sum_{n=0}^{M-1} h(n) e^{-j\omega n}, \quad -\pi < \omega \leqslant \pi$$

现在讨论对 $h(n)$ 和 $H(e^{j\omega})$ 形式的要求以及对 $(M-1)$ 个零点位置的要求,这些要求是由于线性相位约束所施加的。

脉冲响应 $h(n)$

给系统施加一个线性相位约束

$$\angle H(e^{j\omega}) = -\alpha\omega, \quad -\pi < \omega \leqslant \pi$$

其中 α 为恒定相位延迟。从第六章已知,$h(n)$ 必定是对称的。即:

$$h(n) = h(M-1-n), 0 \leqslant n \leqslant (M-1) \text{ 及 } \alpha = \frac{M-1}{2} \tag{7.3}$$

因此 $h(n)$ 关于 α 对称,α 是对称指数。存在两种可能类型的对称:

- M 为奇数:在此情形下,$\alpha = (M-1)/2$ 为整数。脉冲响应如下图所示。

对称脉冲响应: M 为奇数

图 7.2 M 为奇数时的对称脉冲响应

- M 为偶数:在此情形下,$\alpha = (M-1)/2$ 不为整数。脉冲响应如图 7.3 所示。

如果相位响应 $\angle H(e^{j\omega})$ 需要满足条件

$$\angle H(e^{j\omega}) = \beta - \alpha\omega$$

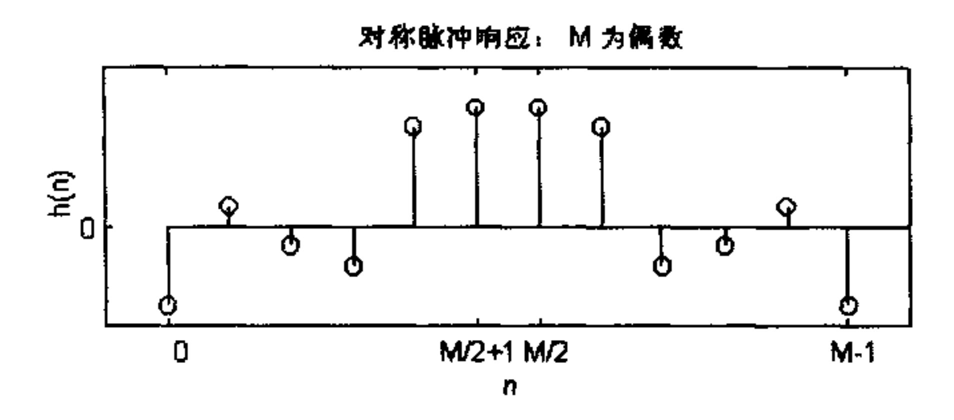


图 7.3　M 为偶数时的对称脉冲响应

则得到第二种“线性相位”FIR 滤波器,它是一条直线,但不通过原点。在这种情况下,α 不是恒定的相位延迟,但是

$$\frac{d\angle H(e^{j\omega})}{d\omega}=-\alpha$$

为常数,它是一个群延迟。因此,α 叫做恒定群延迟。在这种情况下,作为一群,频率以一恒定速率延迟。但是某些频率的延迟大,而其余的则延迟小。对于这种类型的线性相位,可以证明

$$h(n)=-h(M-1-n),0\leqslant n\leqslant(M-1);\ \alpha=\frac{M-1}{2},\beta=\pm\frac{\pi}{2} \tag{7.4}$$

这表明脉冲响应 $h(n)$ 是反对称的。对称指数仍为 $\alpha=(M-1)/2$,我们再次得到两种可能的类型。一种是 M 为奇数,一种是 M 为偶数。

- M 为奇数:在此情形下,$\alpha=(M-1)/2$ 为整数。脉冲响应如图 7.4 所示。

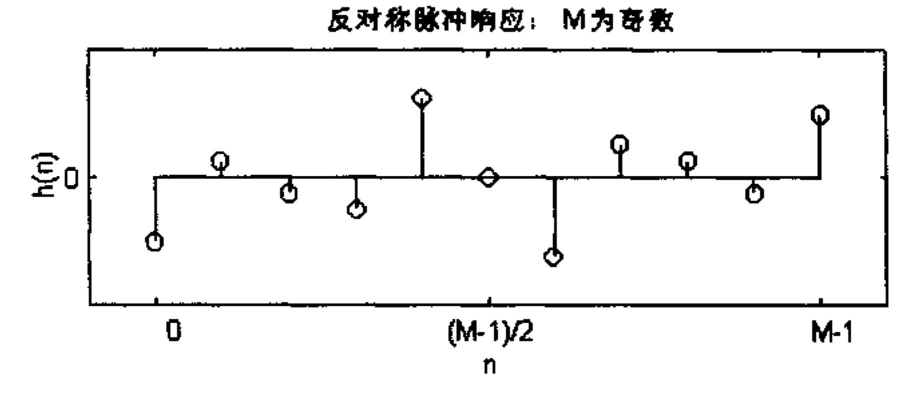


图 7.4　M 为奇数时的反对称脉冲响应

- M 为偶数:在此情形下,$\alpha=(M-1)/2$ 不为整数。脉冲响应如图 7.5 所示。

频率响应 $H(e^{j\omega})$

把 M 为奇数和偶数情形与对称和反对称组合在一起,得到 4 种类型的线性相位 FIR 滤波器。每种情形下的频率响应函数具有特定的表示和形状。为了研究这些响应,把 $H(e^{j\omega})$ 改



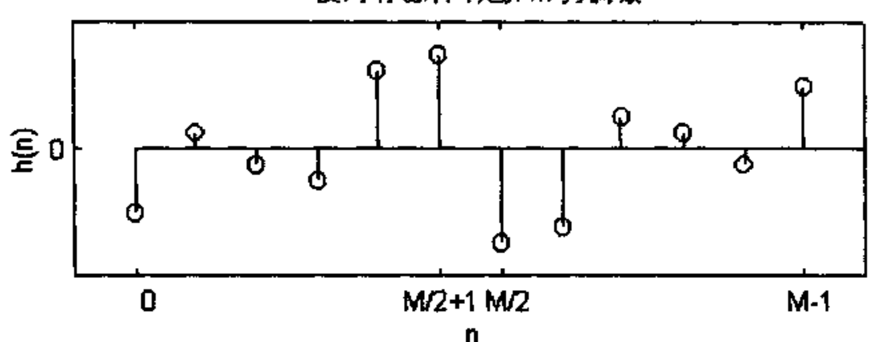


图 7.5　M 为偶数时的反对称脉冲响应

写成

$$H(e^{j\omega}) = H_r(\omega)e^{j(\beta-\alpha\omega)};\beta = \pm\frac{\pi}{2},\alpha = \frac{M-1}{2} \tag{7.5}$$

其中 $H_r(\omega)$是振幅响应函数而不是幅度响应函数。振幅响应是实函数，它既可以为正也可以为负。不像幅度响应永远为正的，与幅度响应有关的相位响应是一个不连续的函数，而与振幅响应有关的相位响应是一个连续的线性函数。下面的例子说明这两种类型之间的差别。

例 7.3　设脉冲响应 $h(n)=\{\underset{\uparrow}{1},1,1\}$，确定并画出频率响应。

解：频率响应函数为：

$$\begin{aligned}H(e^{j\omega}) &= \sum_{0}^{2}h(n)e^{j\omega n} = 1+1e^{-j\omega}+e^{-j2\omega} = \{e^{j\omega}+1+e^{-j\omega}\}e^{-j\omega}\\&= \{1+2\cos\omega\}e^{-j\omega}\end{aligned}$$

由此得到幅度和相位响应为：

$$|H(e^{j\omega})| = |1+2\cos\omega|,0<\omega\leqslant\pi$$

$$\angle H(e^{j\omega}) = \begin{cases}-\omega, & 0<\omega<2\pi/3\\ \pi-\omega, & 2\pi/3<\omega<\pi\end{cases}$$

$\cos\omega$ 既可为正，也可为负。在这种情况下，相位响应是分段线性的。另一方面，振幅和相应的相位响应为：

$$\left.\begin{aligned}H_r(\omega) &= 1+2\cos\omega,\\ \angle H(e^{j\omega}) &= -\omega\end{aligned}\right\} \quad -\pi<\omega\leqslant\pi$$

在这种情况下，相位响应是真正线性的。这些响应如图 7.6 所示。由这个例子可清楚看出幅度和振幅响应（或分段线性和线性相位）之间的差别。

1 型线性相位 FIR 滤波器：对称脉冲响应，M 为奇数

在这种情况下，$\beta=0,\alpha=(M-1)/2$ 为整数，$h(n)=h(M-1-n),0\leqslant n\leqslant M-1$。可以

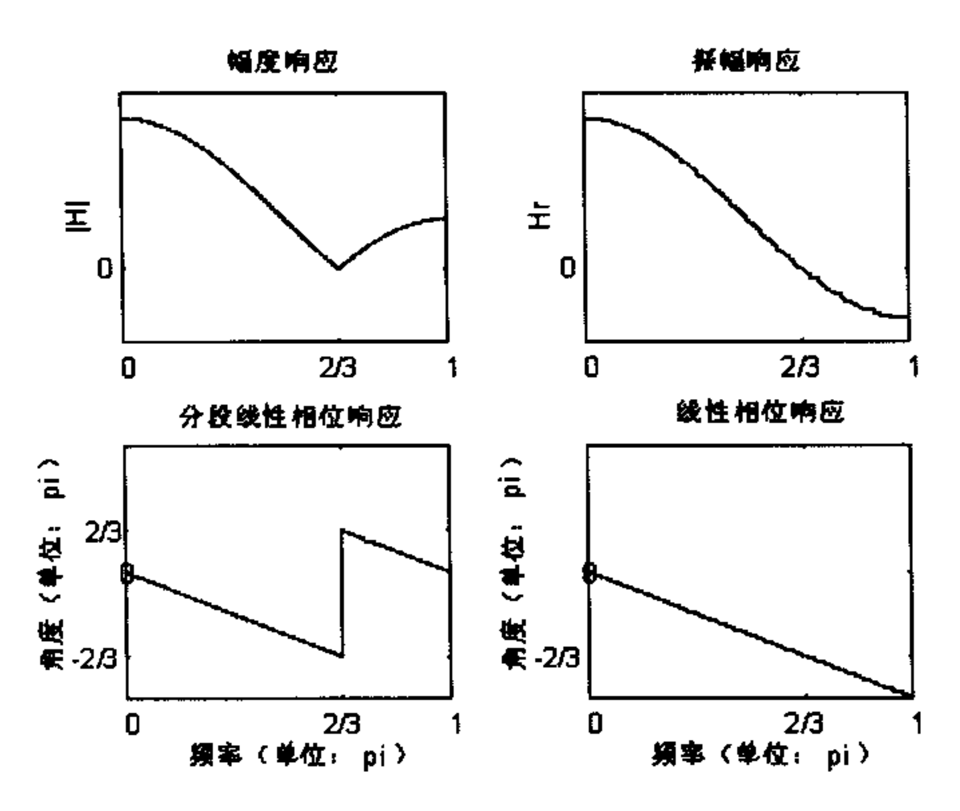


图 7.6　例 7.3 中的频率响应

证明(见习题 7.1)：

$$H(e^{j\omega}) = \left[\sum_{n=0}^{(M-1)/2} a(n)\cos\omega n\right]e^{-j\omega(M-1)/2} \tag{7.6}$$

其中序列 $a(n)$从 $h(n)$得到：

$$\begin{aligned} a(0) &= h\left(\frac{M-1}{2}\right):\text{中间样本} \\ a(n) &= 2h\left(\frac{M-1}{2}-n\right), 1 \leqslant n \leqslant \frac{M-3}{2} \end{aligned} \tag{7.7}$$

比较(7.5)与(7.6),得到

$$H_r(\omega) = \sum_{n=0}^{(M-1)/2} a(n)\cos\omega n \tag{7.8}$$

2 型线性相位 FIR 滤波器:对称脉冲响应,M 为偶数

在这种情况下,$\beta=0, h(n)=h(M-1-n), 0\leqslant n\leqslant M-1$,但 $\alpha=(M-1)/2$ 不为整数,可以证明(见习题 7.2)：

$$H(e^{j\omega}) = \left[\sum_{n=1}^{M/2} b(n)\cos\left\{\omega\left(n-\frac{1}{2}\right)\right\}\right] e^{-j\omega(M-1)/2} \tag{7.9}$$

其中

$$b(n) = 2h\left(\frac{M}{2}-n\right), n = 1,2,\ldots,\frac{M}{2} \tag{7.10}$$

因此

$$H_r(\omega) = \sum_{n=1}^{M/2} b(n)\cos\left\{\omega\left(n-\frac{1}{2}\right)\right\} \tag{7.11}$$

注意:在 $\omega=\pi$ 处,有

$$H_r(\pi) = \sum_{n=1}^{M/2} b(n)\cos\left\{\pi\left(n-\frac{1}{2}\right)\right\} = 0$$

与 $b(n)$或 $h(n)$无关。这种类型(即,对称 $h(n)$, M 为偶数)不能用于高通或带阻滤波器。

3 型线性相位 FIR 滤波器:反对称脉冲响应,M 为奇数

在这种情况下,$\beta=\pi/2, \alpha=(M-1)/2$ 为整数,$h(n)=-h(M-1-n), 0\leqslant n\leqslant M-1$,并且 $h(M-1/2)=0$。可以证明(见习题 7.3):

$$H(e^{j\omega}) = \left[\sum_{n=1}^{(M-1)/2} c(n)\sin\omega n\right] e^{j\left[\frac{\pi}{2}-\left(\frac{M-1}{2}\right)\omega\right]} \tag{7.12}$$

其中

$$c(n) = 2h\left(\frac{M-1}{2}-n\right), n = 1,2,\ldots,\frac{M-1}{2} \tag{7.13}$$

并且

$$H_r(\omega) = \sum_{n=1}^{(M-1)/2} c(n)\sin\omega n \tag{7.14}$$

注意:在 $\omega=0$ 和在 $\omega=\pi$ 处,有 $H_r(\omega)=0$,而与 $c(n)$或 $h(n)$的值无关。而且,$e^{j\pi/2}=j$,这说明 $jH_r(\omega)$是纯虚数。因此,这种类型的滤波器不适用于低通或高通滤波器设计。然而,对于逼近理想数字希尔伯特变换和微分器,它是很有用的。理想希尔伯特变换[19]是一个全通滤波器,它对输入信号产生 90 度的相移,它频繁用于通信系统中的调制。微分器广泛用于模拟和数字系统中对信号求导。

4 型线性相位 FIR 滤波器:反对称脉冲响应,M 为偶数

这种情况与第二种类型相似,有(见习题 7.4)

$$H(e^{j\omega}) = \left[\sum_{n=1}^{M/2} d(n)\sin\left\{\omega\left(n-\frac{1}{2}\right)\right\}\right] e^{j\left[\frac{\pi}{2}-\omega(M-1)/2\right]} \tag{7.15}$$

其中

$$d(n) = 2h\left(\frac{M}{2}-n\right), n = 1,2,\ldots,\frac{M}{2} \tag{7.16}$$

并且

$$H_r(\omega) = \sum_{n=1}^{M/2} d(n)\sin\left\{\omega\left(n-\frac{1}{2}\right)\right\} \tag{7.17}$$

注意:在 $\omega=0$ 处,$H_r(0)=0$ 并且 $e^{j\pi/2}=j$。因此这种类型对设计希尔伯特变换和微分器均适用。

MATLAB 实现

MATLAB 程序 freqz 计算频率响应,但不能从中确定振幅响应。这是因为在 MATLAB 函数中,不存在与 abs 函数功能相当而能找到振幅的函数。但是,很容易写出用来计算每种类型的振幅响应的简单程序。我们提供 4 个函数来完成这个功能。

1. Hr_type1:

```
function [Hr,w,a,L] = Hr_Type1(h);
% 计算 1 型低通滤波器设计的振幅响应 Hr(w)
% ------------------------------
% [Hr,w,a,L] = Hr_Type1(h)
% Hr = 振幅响应
% w = 在[0 pi] 区间内计算 Hr 的 500个频率点
% a = 1 型低通滤波器的系数
% L = Hr 的阶次
% h = 1 型低通滤波器的脉冲响应
%
M = length(h);
L = (M-1)/2;
a = [h(L+1) 2*h(L:-1:1)]; % 1 乘(L+1) 行向量
n = [0:1:L];               % (L+1)乘 1 列向量
w = [0:1:500]'*pi/500;
Hr = cos(w*n)*a';
```

2. Hr _ type2:

```
function [Hr,w,b,L] = Hr_Type2(h);
% 计算 2 型低通滤波器设计的振幅响应 Hr(w)
% ----------------------------
% [Hr,w,b,L] = Hr_Type2(h)
% Hr = 振幅响应
% w = 在[0 pi] 区间内计算 Hr 的频率点
% b = 2 型低通滤波器的系数
% L = Hr 的阶次
% h = 2 型低通滤波器的脉冲响应
%
M = length(h);
L = M/2;
b = 2*[h(L:-1:1)];
n = [1:1:L]; n = n-0.5;
w = [0:1:500]'*pi/500;
Hr = cos(w*n)*b';
```

3. Hr _ type3:

```
function [Hr,w,c,L] = Hr_Type3(h);
% 计算 3 型低通滤波器设计的振幅响应 Hr(w)
% -------------------------
% [Hr,w,c,L] = Hr_Type3(h)
% Hr = 振幅响应
% w = 在[0 pi] 区间内计算 Hr 的频率点
% c = 3 型低通滤波器的系数
% L = Hr 的阶次
% h = 3 型低通滤波器的脉冲响应
%
M = length(h);
L = (M-1)/2;
c = [2*h(L+1:-1:1)];
n = [0:1:L];
w = [0:1:500]'*pi/500;
Hr = sin(w*n)*c';
```

4. Hr _ type4:

```
function [Hr,w,d,L] = Hr_Type4(h);
% 计算 4 型低通滤波器设计的振幅响应 Hr(w)
% -------------------------
```

```
% [Hr,w,d,L] = Hr_Type4(h)
% Hr = 振幅响应
% w = 在[0 pi] 区间内计算 Hr 的 500 个频率点
% d = 4 型低通滤波器的系数
% L = d 的阶次
% h = 4 型低通滤波器的脉冲响应
%
M = length(h);
L = M/2;
d = 2*[h(L:-1:1)];
n = [1:1:L]; n = n-0.5;
w = [0:1:500]'*pi/500;
Hr = sin(w*n)*d';
```

这四个函数可组合成一个函数。此函数首先确定线性相位的类型,然后实现相应的振幅响应。这会在习题 7.5 中研究。这些函数的用途在例 7.4 至 7.7 中说明。

零点位置

回想在原点有($M-1$)重极点及在 z 平面上有($M-1$)个零点的 FIR 滤波器,对于线性相位 FIR 滤波器来讲,由于 $h(n)$上具有的对称性约束,这些零点也具有一定的对称性。可以证明(见文献[19]和习题 7.6)如果 $H(z)$在

$$z = z_1 = re^{j\theta}$$

上有一零点,那么由于是线性相位,在

$$z = \frac{1}{z_1} = \frac{1}{r}e^{-j\theta}$$

处必定存在一零点。对于一个实数滤波器,如果 z_1 为复数,则在 $z_1^* = re^{-j\theta}$处有一个共轭零点,这意味着在 $1/z_1^* = (1/r)e^{j\theta}$处也有一零点,因此,一般的零点构象是一个四件套,

$$re^{j\theta}, \qquad \frac{1}{r}e^{j\theta}, \qquad re^{-j\theta}, \qquad 以及\frac{1}{r}e^{-j\theta}$$

如图 7.7 所示。显然,如果 $r=1$,则 $1/r=1$,因此零点位于单位圆上,且成对出现。

$$e^{j\theta}和\ e^{-j\theta}$$

如果 $\theta=0$ 或 $\theta=\pi$,则零点在实轴上成对出现。

$$r \quad 和 \quad \frac{1}{r}$$

最后,如果 $r=1$ 并且 $\theta=0$ 或 $\theta=\pi$,则零点位于 $z=1$ 或 $z=-1$ 处。利用这些对称性,可用线性相位子系统实现级联形式

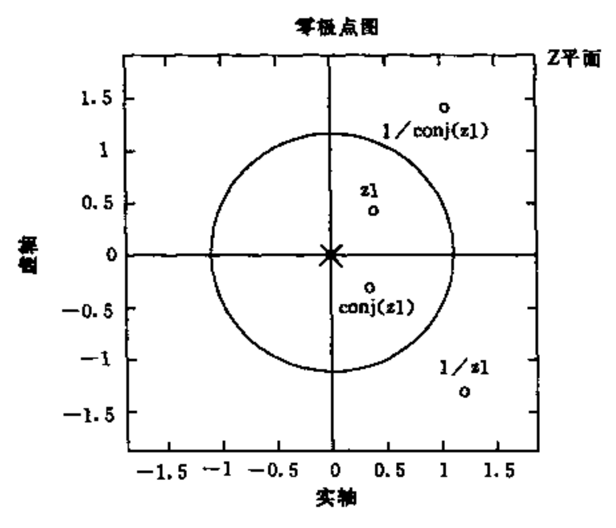


图 7.7　零点构象图

在下面的例子中,将说明上面讨论的线性相位 FIR 滤波器性质。

例 7.4　设 $h(n)=\{-\underset{\uparrow}{4},1,-1,-2,5,6,5,-2,-1,1,-4\}$,求出振幅响应 $H_r(\omega)$ 以及 $H(z)$ 的零点位置。

解: 由于 $M=11$,为奇数,$h(n)$ 关于 $\alpha=(11-1)/2=5$ 对称,这是一个 1 型线性相位 FIR 滤波器。由(7.7)得:

$$a(0)=h(\alpha)=h(5)=6,\ a(1)=2h(5-1)=10,\ a(2)=2h(5-2)=-4$$
$$a(3)=2h(5-3)=-2,\ a(4)=2h(5-4)=2,\ a(5)=2h(5-5)=-8$$

由(7.8)得:

$$\begin{aligned}H_r(\omega)&=a(0)+a(1)\cos\omega+a(2)\cos2\omega+a(3)\cos3\omega+a(4)\cos4\omega+a(5)\cos5\omega\\&=6+10\cos\omega-4\cos2\omega-2\cos3\omega+2\cos4\omega-8\cos5\omega\end{aligned}$$

MATLAB 程序

```
>> h = [-4,1,-1,-2,5,6,5,-2,-1,1,-4];
>> M = length(h); n = 0:M-1;
>> [Hr,w,a,L] = Hr_Type1(h);
>> a,L
a = 6    10    -4    -2    2    -8
L = 5
>> amax = max(a)+1; amin = min(a)-1;
>> subplot(2,2,1); stem(n,h); axis([-1 2*L+1 amin amax])
```

```
>> xlabel('n');ylabel('h(n)'); title('脉冲响应')
>> subplot(2,2,3); stem(0:L,a); axis([-1 2*L+1 amin amax])
>> xlabel('n');ylabel('a(n)'); title('a(n) 系数')
>> subplot(2,2,2);plot(w/pi,Hr);grid
>> xlabel('频率(单位:pi)'); ylabel('Hr')
>> title('1 型振幅响应')
>> subplot(2,2,4);pzplotz(h,1)
```

所得响应图和零点位置如图 7.8 所示。从图可观察到无论 $\omega=0$ 或 $\omega=\pi$，$H_r(\omega)$均没有约束。其零点构象为一个零点 4 件套和三个零点对。

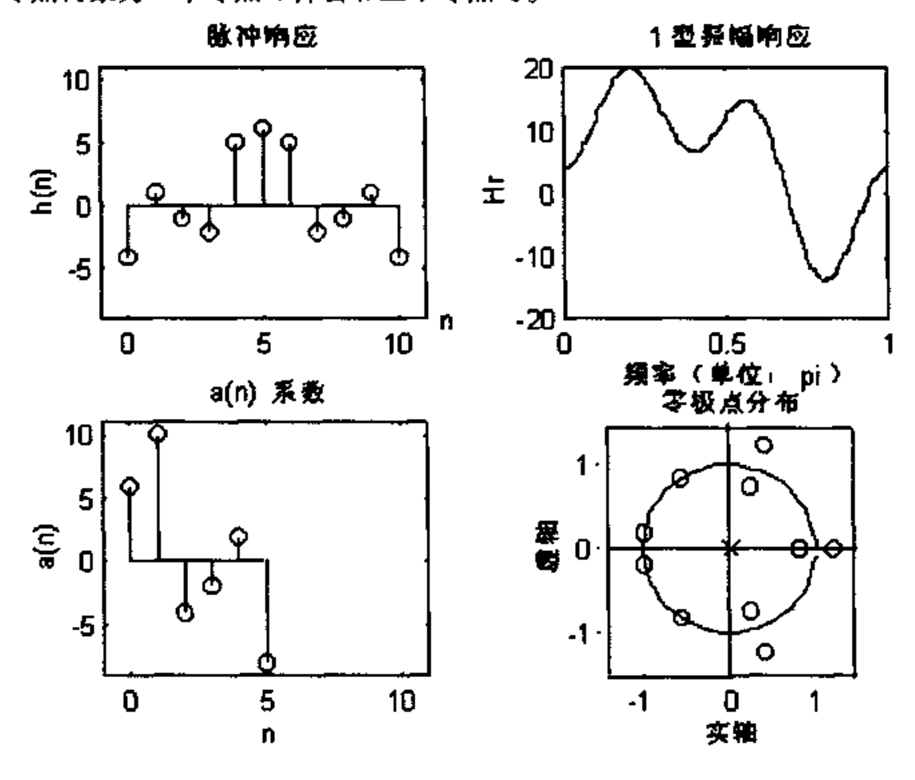


图 7.8　例 7.4 中的图

例 7.5　设 $h(n)=\{-\underset{\uparrow}{4},1,-1,-2,5,6,6,5,-2,-1,1,-4\}$，确定振幅响应 $H_r(\omega)$以及 $H(z)$的零点位置。

解：由于 $M=12$，$h(n)$关于 $\alpha=(12-1)/2=5.5$ 对称，这是一个 2 型线性相位 FIR 滤波器。由(7.10)得：

$$b(1)=2h\left(\frac{12}{2}-1\right)=12, b(2)=2h\left(\frac{12}{2}-2\right)=10, b(3)=2h\left(\frac{12}{2}-3\right)=-4$$

$$b(4)=2h\left(\frac{12}{2}-4\right)=-2, b(5)=2h\left(\frac{12}{2}-5\right)=2, b(6)=2h\left(\frac{12}{2}-6\right)=-8$$

由(7.11)得：

$$
\begin{aligned}
H_r(\omega) &= b(1)\cos\left[\omega\left(1-\frac{1}{2}\right)\right] + b(2)\cos\left[\omega\left(2-\frac{1}{2}\right)\right] + b(3)\cos\left[\omega\left(3-\frac{1}{2}\right)\right] \\
&\quad + b(4)\cos\left[\omega\left(4-\frac{1}{2}\right)\right] + b(5)\cos\left[\omega\left(5-\frac{1}{2}\right)\right] + b(6)\cos\left[\omega\left(6-\frac{1}{2}\right)\right] \\
&= 12\cos\left(\frac{\omega}{2}\right) + 10\cos\left(\frac{3\omega}{2}\right) - 4\cos\left(\frac{5\omega}{2}\right) - 2\cos\left(\frac{7\omega}{2}\right) \\
&\quad + 2\cos\left(\frac{9\omega}{2}\right) - 8\cos\left(\frac{11\omega}{2}\right)
\end{aligned}
$$

MATLAB 程序

```
>> h = [-4,1,-1,-2,5,6,6,5,-2,-1,1,-4];
>> M = length(h); n = 0:M-1;
>> [Hr,w,b,L] = Hr_Type2(h);
>> b,L
b =  12    10    -4    -2    2    -8
L = 6
>> bmax = max(b)+1; bmin = min(b)-1;
>> subplot(2,2,1); stem(n,h); axis([-1 2*L+1 bmin bmax])
>> xlabel('n');ylabel('h(n)'); title('脉冲响应')
>> subplot(2,2,3); stem(1:L,b); axis([-1 2*L+1 bmin bmax])
>> xlabel('n'); ylabel('b(n)'); title('b(n) 系数')
>> subplot(2,2,2);plot(w/pi,Hr);grid
>> xlabel('频率(单位:pi)'); ylabel('Hr')
>> title('2 型振幅响应')
>> subplot(2,2,4);pzplotz(h,1)
```

所得响应图和零点位置如图 7.9 所示。从图可观察到 $\omega=\pi$ 时，$H_r(\omega)$为零。其零点构象为一个零点 4 件套，三个零点对以及在 $\omega=\pi$ 处有一个零点，这与预想的相同。

例 7.6 设 $h(n)=\{\underset{\uparrow}{-4},1,-1,-2,5,0,-5,2,1,-1,4\}$，确定振幅响应 $H_r(\omega)$以及 $H(z)$的零点位置。

解：由于 $M=11$，为奇数，$h(n)$关于 $\alpha=(11-1)/2=5$ 对称，这是一个 3 型线性相位 FIR 滤波器。由(7.13)得：

$$
\begin{aligned}
c(0) &= h(\alpha) = h(5) = 0, c(1) = 2h(5-1) = 10, c(2) = 2h(2-2) = -4, \\
c(3) &= 2h(5-3) = -2, c(4) = 2h(5-4) = 2, c(5) = 2h(5-5) = -8
\end{aligned}
$$

由(7.14)得：

$$
\begin{aligned}
H_r(\omega) &= c(0) + c(1)\sin\omega + c(2)\sin2\omega + c(3)\sin3\omega + c(4)\sin4\omega + c(5)\sin5\omega \\
&= 0 + 10\sin\omega - 4\sin2\omega - 2\sin3\omega + 2\sin4\omega - 8\sin5\omega
\end{aligned}
$$

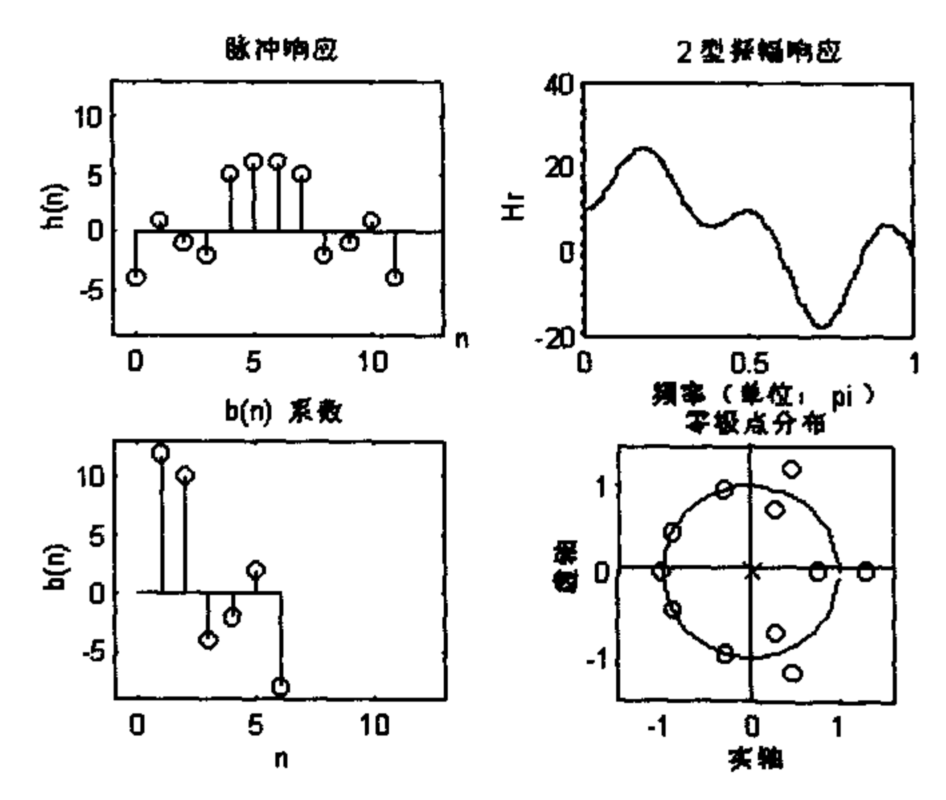


图 7.9 例 7.5 中的图

MATLAB 程序

```
>> h = [-4,1,-1,-2,5,0,-5,2,1,-1,4];
>> M = length(h); n = 0:M-1;
>> [Hr,w,c,L] = Hr_Type3(h);
>> c,L
c = 0    10    -4    -2      2      -8
L = 5
>> cmax = max(c)+1; cmin = min(c)-1;
>> subplot(2,2,1); stem(n,h); axis([-1 2*L+1 cmin cmax])
>> xlabel('n'); ylabel('h(n)'); title('脉冲响应')
>> subplot(2,2,3); stem(0:L,c); axis([-1 2*L+1 cmin cmax])
>> xlabel('n');ylabel('c(n)'); title('c(n)系数')
>> subplot(2,2,2);plot(w/pi,Hr);grid
>> xlabel('频率(单位:pi)'); ylabel('Hr')
>> title('3 型振幅响应')
>> subplot(2,2,4);pzplotz(h,1)
```

所得响应图和零点位置如图 7.10 所示。从图可观察到 $\omega=0$ 及 $\omega=\pi$ 时，$H_r(\omega)$为零。其零点构象为一个零点 4 件套，两个零点对以及在 $\omega=0$ 及 $\omega=\pi$ 处有零点，这与预想的相同。

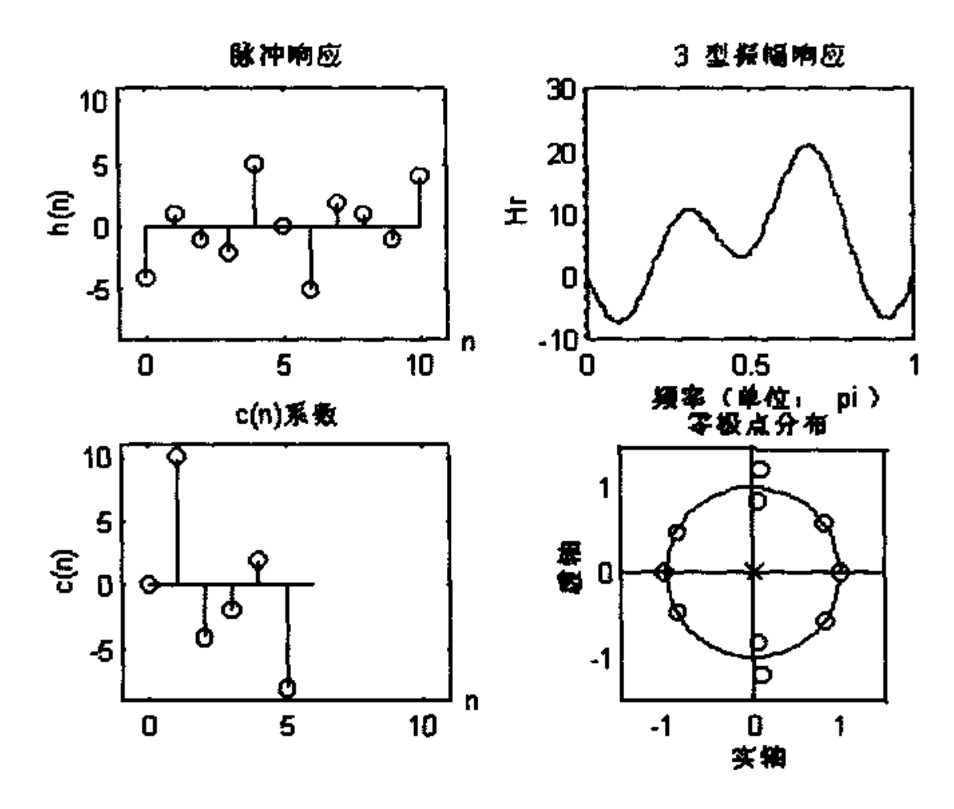


图 7.10 例 7.6 中的图

例 7.7 设 $h(n)=\{-\underset{\uparrow}{4},1,-1,-2,5,6,-6,-5,2,1,-1,4\}$，确定振幅响应 $H_r(\omega)$ 以及 $H(z)$ 的零点位置。

解：由于 $M=12$，$h(n)$ 关于 $\alpha=(12-1)/2=5.5$ 反对称，这是一个 4 型线性相位 FIR 滤波器。由(7.16)得：

$$d(1)=2h\left(\frac{12}{2}-1\right)=12,d(2)=2h\left(\frac{12}{2}-2\right)=10,d(3)=2h\left(\frac{12}{2}-3\right)=-4$$

$$d(4)=2h\left(\frac{12}{2}-4\right)=-2,d(5)=2h\left(\frac{12}{2}-5\right)=2,d(6)=2h\left(\frac{12}{2}-6\right)=-8$$

由(7.17)得：

$$\begin{aligned}H_r(\omega)&=d(1)\sin\left[\omega\left(1-\frac{1}{2}\right)\right]+d(2)\sin\left[\omega\left(2-\frac{1}{2}\right)\right]+d(3)\sin\left[\omega\left(3-\frac{1}{2}\right)\right]\\&\quad+d(4)\sin\left[\omega\left(4-\frac{1}{2}\right)\right]+d(5)\sin\left[\omega\left(5-\frac{1}{2}\right)\right]+d(6)\sin\left[\omega\left(6-\frac{1}{2}\right)\right]\\&=12\sin\left(\frac{\omega}{2}\right)+10\sin\left(\frac{3\omega}{2}\right)-4\sin\left(\frac{5\omega}{2}\right)-2\sin\left(\frac{7\omega}{2}\right)+2\sin\left(\frac{9\omega}{2}\right)-8\sin\left(\frac{11\omega}{2}\right)\end{aligned}$$

MATLAB 程序——————————

```
>> h = [-4,1,-1,-2,5,6,-6,-5,2,1,-1,4];
>> M = length(h); n = 0:M-1;
>> [Hr,w,d,L] = Hr_Type4(h);
```

```
>>d,L
d = 12     10     -4     -2     2     -8
L =  6
>> dmax = max(d)+1; dmin = min(d)-1;
>> subplot(2,2,1); stem(n,h); axis([-1 2*L+1 dmin dmax])
>> xlabel('n'); ylabel('h(n)'); title('脉冲响应')
>> subplot(2,2,3); stem(1:L,d); axis([-1 2*L+1 dmin dmax])
>> xlabel('n'); ylabel('d(n)'); title('d(n) 系数')
>> subplot(2,2,2);plot(w/pi,Hr);grid
>> xlabel('频率(单位:pi)'); ylabel('Hr')
>> title('4 型振幅响应')
>> subplot(2,2,4);pzplotz(h,1)
```

所得响应图和零点位置如图 7.11 所示。从图可观察到 $\omega=0$ 时，$H_r(\omega)$为零。其零点构象为一个零点 4 件套，三个零点对以及在 $\omega=0$ 处有一个零点，这与预想的相同。

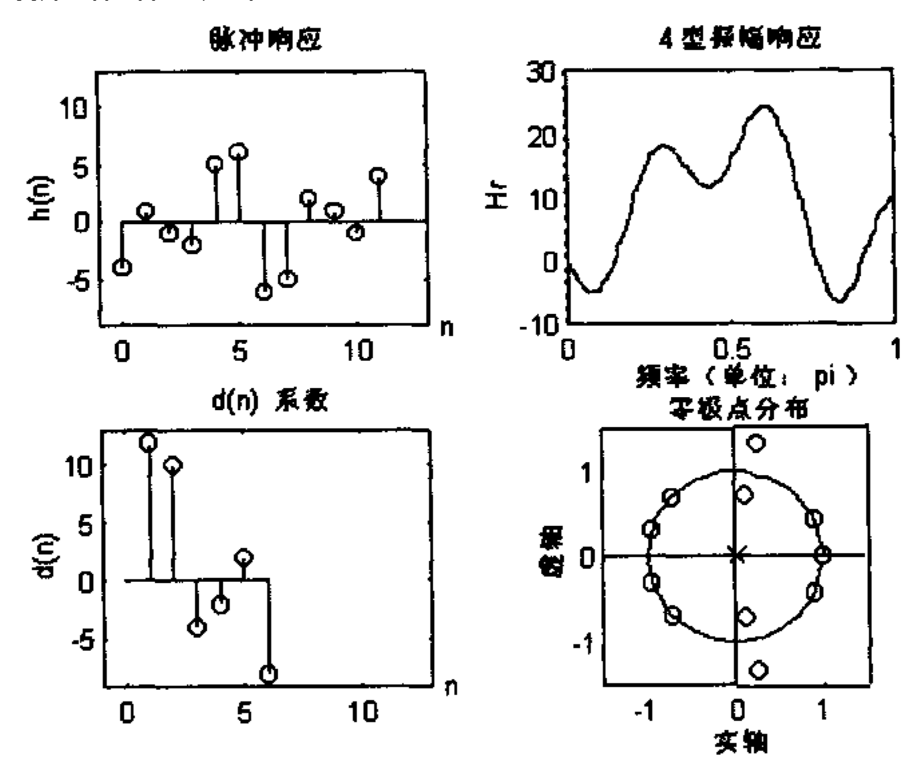


图 7.11　例 7.7 中的图

窗设计技术

窗设计的基本思想为，首先选择一个适当的理想选频滤波器（它总是具有一个非因果，无限持续时间脉冲响应），然后截取（或加窗）它的脉冲响应得到线性相位和因果 FIR 滤波器。因此这种方法的重点是选择一个合适的窗函数和理想滤波器。我们用 $H_d(e^{j\omega})$表示理想的选

频滤波器，它在通带上具有单位增益和线性相位，在阻带上具有零响应。一个带宽 $\omega_c < \pi$ 的低通滤波器由下式给定：

$$H_d(e^{j\omega}) = \begin{cases} 1 \cdot e^{-j\alpha\omega}, & |\omega| \leq \omega_c \\ 0, & \omega_c < |\omega| \leq \pi \end{cases} \tag{7.18}$$

其中 ω_c 也叫做截止频率，α 为采样延迟（注意根据 DTFT 性质，$e^{-j\alpha\omega}$意味着在 n 正方向移位或延迟）。这个滤波器的脉冲响应具有无限持续时间，它由下式给定：

$$\begin{aligned} h_d(n) = F^{-1}[H_d(e^{j\omega})] &= \frac{1}{2\pi}\int_{-\pi}^{\pi} H_d(e^{j\omega})e^{j\omega n}d\omega \\ &= \frac{1}{2\pi}\int_{-\omega_c}^{\omega_c} 1 \cdot e^{-j\alpha\omega}e^{j\omega n}d\omega \\ &= \frac{\sin[\omega_c(n-\alpha)]}{\pi(n-\alpha)} \end{aligned}$$

注意 $h_d(n)$关于 α 对称，这一点对线性相位滤波器是有用的。

为了从 $h_d(n)$得到一个 FIR 滤波器，必须同时在两边截取 $h_d(n)$。而要得到一个因果的线性相位滤波器，它的 $h(n)$长度为 M，必须有：

$$h(n) = \begin{cases} h_d(n), & 0 \leq n \leq M-1 \\ 0, & \text{其它} \end{cases} \qquad \alpha = \frac{M-1}{2} \tag{7.20}$$

这种操作叫做"加窗"，一般地讲，$h(n)$可看作是 $h_d(n)$与窗函数 $w(n)$的乘积：

$$h(n) = h_d(n)w(n) \tag{7.21}$$

其中

$$w(n) = \begin{cases} \text{在 } 0 \leq n \leq M-1 \text{ 上，关于 } \alpha \text{ 对称的一个函数} \\ \text{其他 } n \text{ 值处，} 0 \end{cases}$$

根据上述 $w(n)$的不同定义，可以得到不同的窗结构。例如，在(7.20)中，

$$w(n) = \begin{cases} 1, & 0 \leq n \leq M-1 \\ 0, & \text{其他} \end{cases} = R_M(n)$$

为原先定义的矩形窗。

在频域中，因果 FIR 滤波器响应 $H(e^{j\omega})$由 $H_d(e^{j\omega})$和窗响应 $W(e^{j\omega})$的周期卷积得到，即：

$$H(e^{j\omega}) = H_d(e^{j\omega}) \circledast W(e^{j\omega}) = \frac{1}{2\pi}\int_{-\pi}^{\pi} W(e^{j\lambda}) H_d(e^{j(\omega-\lambda)})\, d\lambda \tag{7.22}$$

图 7.12 给出一个典型的窗响应，从中可以得出以下结论：

1. 由于窗函数 $w(n)$ 有一个等于 M 的有限长度，它的响应有一个主瓣，其宽度与 $1/M$ 成正比，还有高度较小的旁瓣。

2. 周期卷积(7.22)使理想响应 $H_d(e^{j\omega})$ 变得轮廓模糊。

3. 主瓣在 $H(e^{j\omega})$ 中产生一个过渡带，主瓣的宽度决定过渡带的宽度，即与 $1/M$ 成正比。主瓣越宽，则过渡带越宽。

4. 旁瓣产生波动，它在通带和阻带中具有类似的形状。

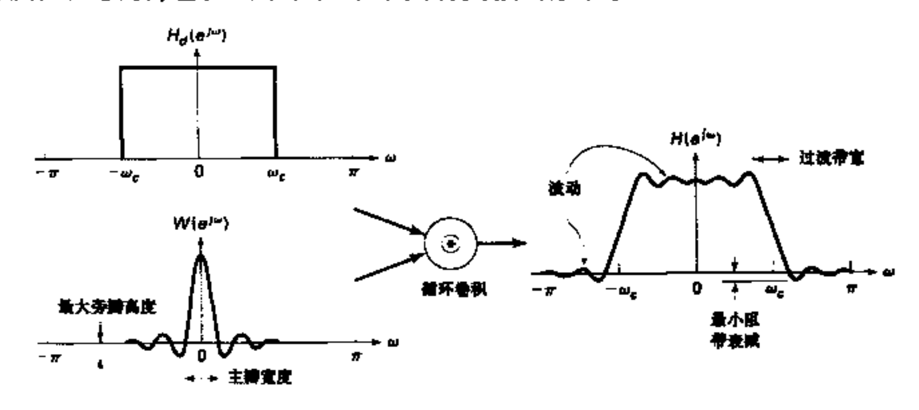


图 7.12 频域中的加窗运算

窗函数设计的基本思想

根据给定的滤波器技术指标，选择滤波器长度 M 和窗函数 $w(n)$，使其具有最窄宽度的主瓣和最小的旁瓣。

根据上面的结论 4，我们注意到，不能独立给出通带容限 δ_1 和阻带容限 δ_2。一般来讲，设 $\delta_1 = \delta_2$，因此只考虑 δ_2。在这里简要讨论常用的窗函数，同时用矩形窗作为一个例子研究这些窗在频域中的特性。

矩形窗

这是一种最简单的窗函数，但从阻带衰减的角度看，其性能最差。它的定义为：

$$\omega(n) = \begin{cases} 1, & 0 \leqslant n \leqslant M-1 \\ 0 & \text{其它} \end{cases} \tag{7.23}$$

它的频率响应函数为：

$$W(e^{j\omega}) = \left[\frac{\sin\left(\frac{\omega M}{2}\right)}{\sin\left(\frac{\omega}{2}\right)}\right] e^{-j\omega\frac{M-1}{2}} \Rightarrow W_r(\omega) = \frac{\sin\left(\frac{\omega M}{2}\right)}{\sin\left(\frac{\omega}{2}\right)}$$

这是一个振幅响应，根据(7.22)，真正的振幅响应 $H_r(\omega)$为：

$$H_r(\omega) \simeq \frac{1}{2\pi}\int_{-\pi}^{\omega+\omega_c} W_r(\lambda)\,d\lambda = \frac{1}{2\pi}\int_{-\pi}^{\omega+\omega_c} \frac{\sin\left(\frac{\omega M}{2}\right)}{\sin\left(\frac{\omega}{2}\right)}\,d\lambda\,,\ M \gg 1 \tag{7.24}$$

由此式可看出，在对过渡带和阻带衰减进行精确分析的过程中，对窗振幅响应进行连续积分(或累积振幅响应)是很必要的。图 7.13 给出矩形窗函数 $w(n)$，它的振幅响应 $W_r(\omega)$，以及以 dB 为单位的振幅响应和累积振幅响应(7.24)。观察 7.13 图可得出一些结论。

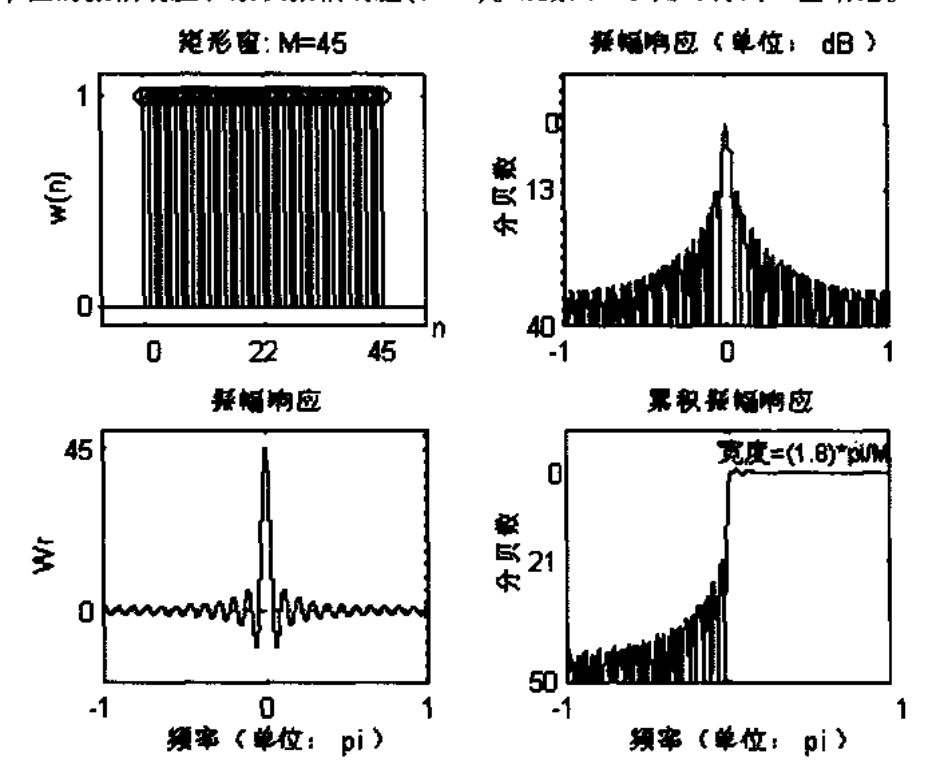


图 7.13 矩形窗 $M=45$

1. 振幅响应 $W_r(\omega)$在 $\omega=\omega_1$ 处具有第一个零点，

$$\frac{\omega_1 M}{2} = \pi \qquad \text{or} \qquad \omega_1 = \frac{2\pi}{M}$$

因而主瓣的宽度为 $2\omega_1 = 4\pi/M$，所以过渡带宽也近似为 $4\pi/M$。

2. 大约在 $\omega = 3\pi/M$ 处，出现第一个旁瓣(它也是主旁瓣)，其幅度为：

$$\left|W_r\left(\omega = \frac{3\pi}{M}\right)\right| = \left|\frac{\sin\left(\frac{3\pi}{2}\right)}{\sin\left(\frac{3\pi}{2M}\right)}\right| \simeq \frac{2M}{3\pi} \qquad M \gg 1 \text{ 时}$$

将它与振幅为 M 的主瓣比较，则这两者之比为：

$$\frac{2}{3\pi} = 21.22\% \equiv 13 \text{ dB}$$

3. 累积振幅响应的第一个旁瓣为 21dB，这个 21dB 的阻带衰减与窗长度 M 无关。

4. 根据最小阻带衰减，可以精确计算过渡带宽。在图 7.13 中的累积振幅响应中已说明了这一点。所计算的精确过渡带宽为：

$$\omega_s - \omega_p = \frac{1.8\pi}{M}$$

它大约是近似带宽 $4\pi/M$ 的一半。

显然，在时域中，加矩形窗是一个简单的运算，同时在频域中，也易于分析。但是，存在两个问题。第一，21dB 的最小阻带衰减在实际应用中是不够的。第二，由于矩形窗对无限长度 $h_d(n)$ 直接截取，因此存在吉布斯(Gibbs)现象。如果增大 M，则每个旁瓣的宽度减小，但每个瓣下的面积保持不变，而且最小阻带衰减仍为 21dB。这意味着，所有的波动集中在带的边沿，如图 7.14 所示。

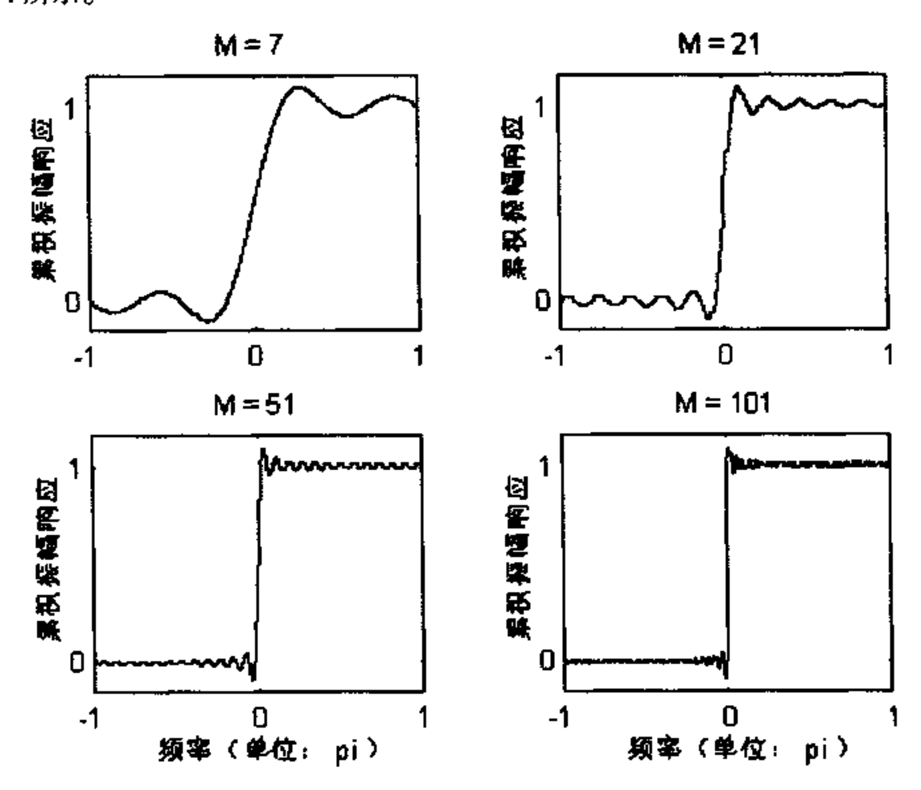


图 7.14 吉布斯现象 $M = 45$

既然矩形窗在许多实际应用中是不适用的，因此我们要考虑其他的窗函数，这些窗大都以最早提出它们的人的名字来命名。由于对这些窗函数可作与矩形窗类似的分析，因此，这里只给出结果。

巴特利特(BARTLETT)窗

由于矩形窗从 0 到 1(或 1 到 0)有一个突跳的过渡带，这造成了吉布斯现象。巴特利特(Bartlett)提出了一种逐渐过渡的三角窗形式，为：

$$
w(n)=\begin{cases}\dfrac{2n}{M-1}, & 0\leqslant n\leqslant\dfrac{M-1}{2}\\[2mm] 2-\dfrac{2n}{M-1}, & \dfrac{M-1}{2}\leqslant n\leqslant M-1\\[2mm] 0, & \text{其它}\end{cases}\tag{7.25}
$$

这种窗结构及其频域响应如图 7.15 所示。

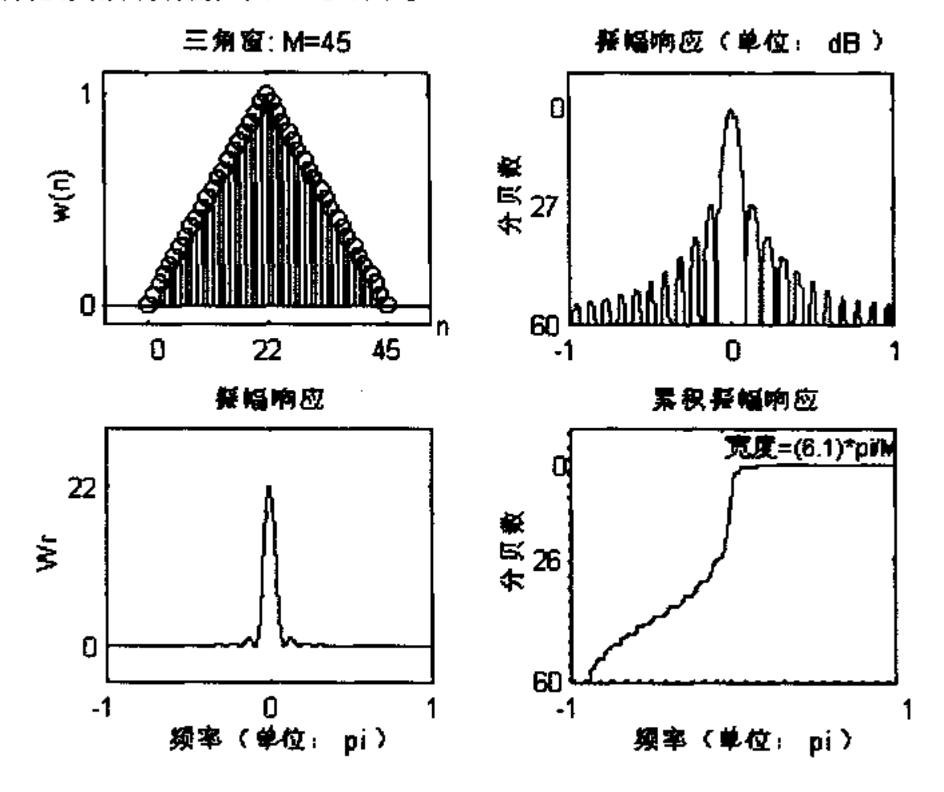


图 7.15　巴特利特(三角)窗　$M=45$

汉宁(HANNING)窗

它的形式是一个凸起的余弦,为:

$$
w(n)=\begin{cases}0.5\left[1-\cos\left(\dfrac{2\pi n}{M-1}\right)\right], 0\leqslant n\leqslant M-1\\[2mm] 0, \quad \text{其它}\end{cases}\tag{7.26}
$$

这种窗结构及其频域响应如图 7.16 所示。

哈明(HAMMING)窗

这种窗除了有少许间断点外,其他与汉宁窗类似:

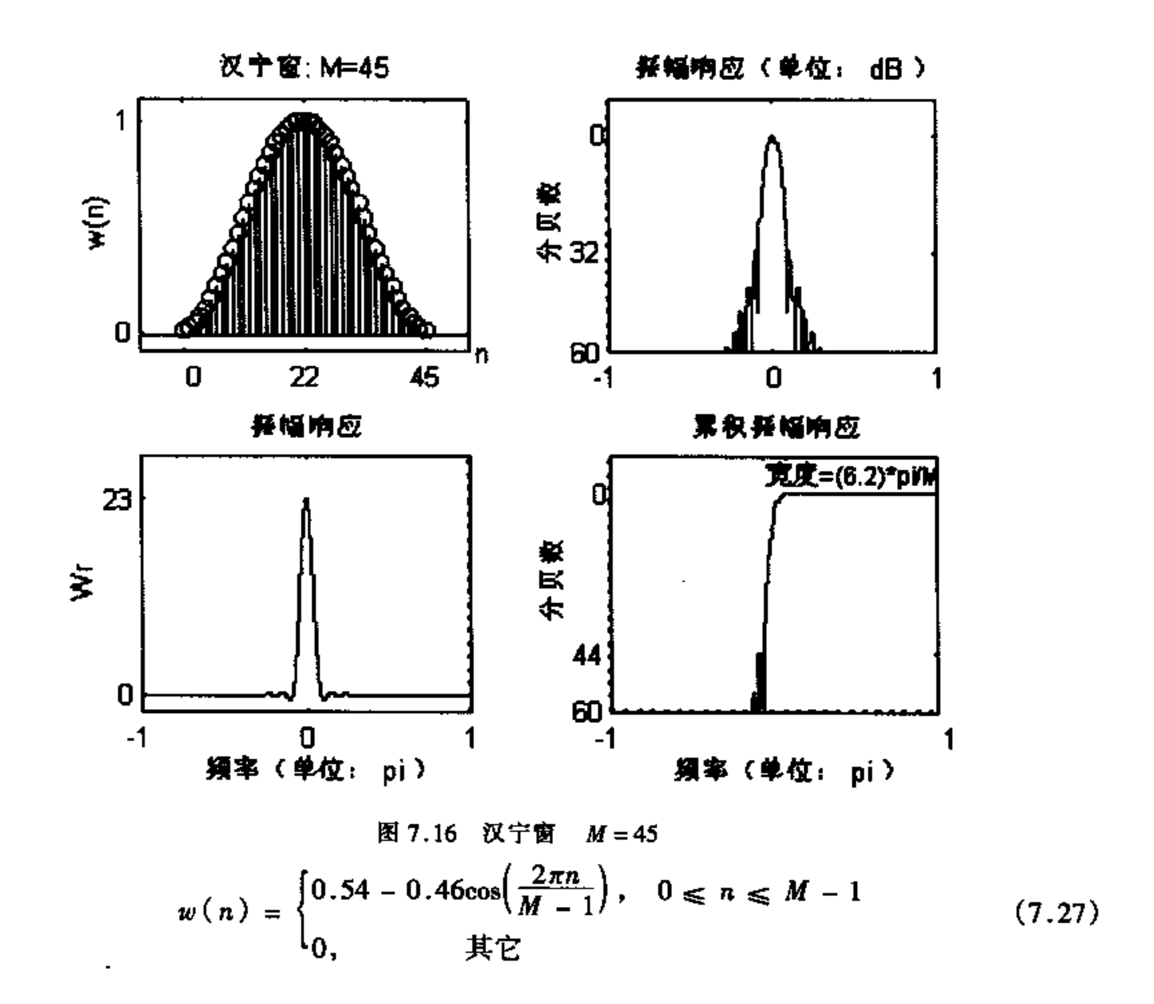


图 7.16 汉宁窗 $M=45$

$$w(n)=\begin{cases}0.54-0.46\cos\left(\dfrac{2\pi n}{M-1}\right), & 0\leqslant n\leqslant M-1\\ 0, & \text{其它}\end{cases}\tag{7.27}$$

这种窗结构及其频域响应如图 7.17 所示。

布莱克曼(BLACKMAN)窗

同样地,这种窗与前两种窗类似,但它包含了二次谐波形式:

$$w(n)=\begin{cases}0.42-0.5\cos\left(\dfrac{2\pi n}{M-1}\right)+0.08\cos\left(\dfrac{4\pi n}{M-1}\right), & 0\leqslant n\leqslant M-1\\ 0, & \text{其它}\end{cases}\tag{7.28}$$

这种窗结构及其频域响应如图 7.18 所示。

表 7.1 归纳了各种窗函数的特性,在表中提供了过渡带宽(作为 M 的函数)和最小阻带衰减的 dB 数。并且给出了近似的和精确的过渡带宽。注意,随着表的下移,过渡带宽和阻带衰减增大。对许多应用来讲,哈明窗似乎是最佳选择。

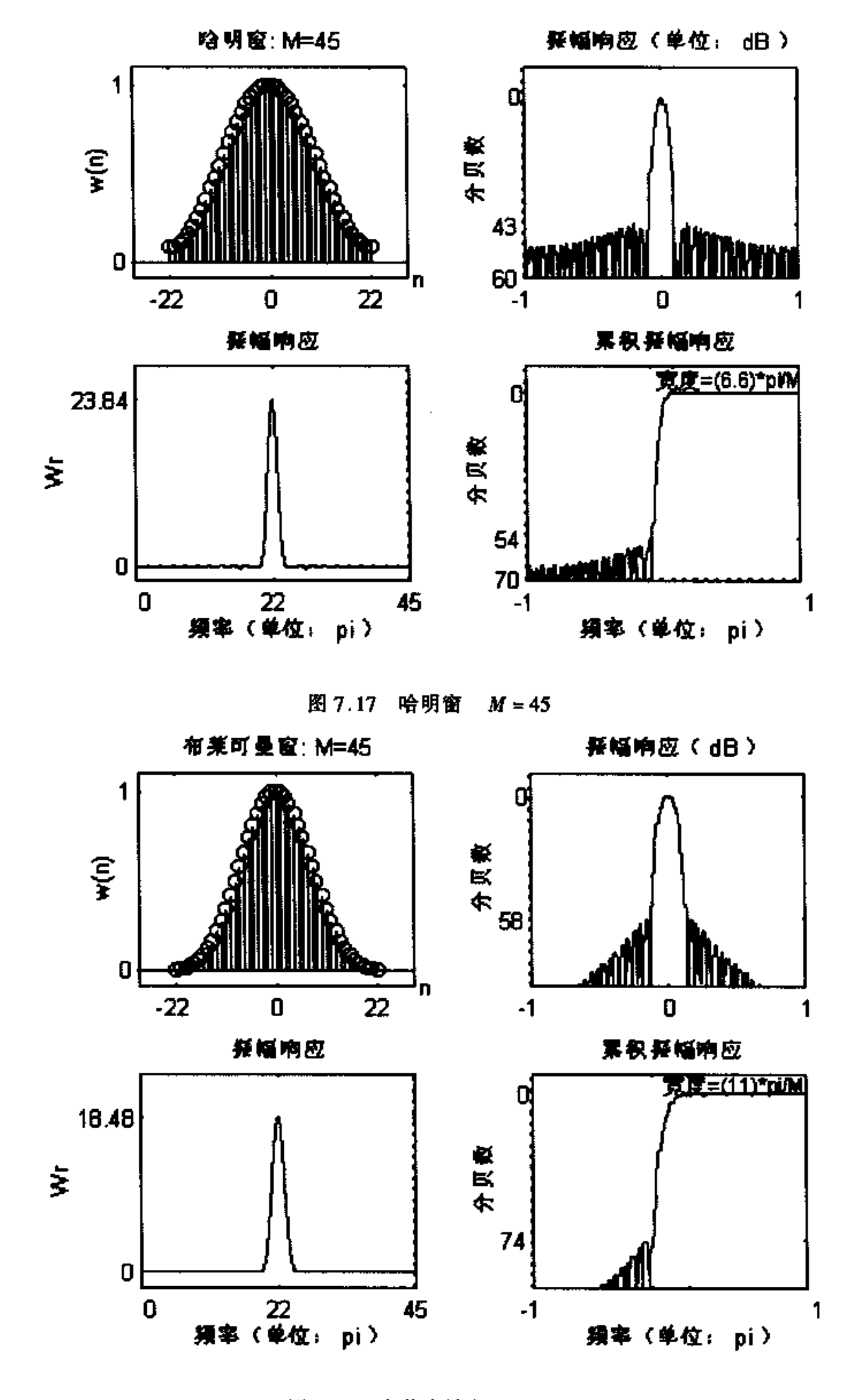


图 7.17 哈明窗 $M=45$

图 7.18 布莱克曼窗 $M=45$

表 7.1　常用窗函数特性一览

窗函数名称	近似过渡带宽	精确过渡带宽	最小阻带衰减
矩形	$4\pi/M$	$1.8\pi/M$	21dB
巴特利特	$8\pi/M$	$6.1\pi/M$	25dB
汉宁	$8\pi/M$	$6.2\pi/M$	44dB
哈明	$8\pi/M$	$6.6\pi/M$	53dB
布莱克曼	$12\pi/M$	$11\pi/M$	74dB

凯泽(KAISER)窗

这是最有用且最优的窗结构之一,对于给定的阻带衰减,它提供了最大的主瓣宽度,也就是最陡的过渡带。就这点而言,它是最优的。窗函数由 J.F.Kaiser 提出,由下式给出:

$$w(n)=\frac{I_0\left[\beta\sqrt{1-\left(1-\frac{2n}{M-1}\right)^2}\right]}{I_0[\beta]},\quad 0\leqslant n\leqslant M-1 \tag{7.29}$$

其中 $I_0[.]$是修正过的零阶贝塞尔(Bessel)函数,β 为依赖于 M 的参数,选择 M 可产生各种过渡带宽和接近最优的阻带衰减。对于相同的 M,此窗可以提供不同的过渡带宽,这是其他窗做不到的。例如

- 如果 $\beta=5.658$,则过渡带宽等于 $7.8\pi/M$,最小阻带衰减为 60dB,如图 7.19 所示。
- 如果 $\beta=4.538$,则过渡带宽等于 $5.8\pi/M$,最小阻带衰减为 50dB。

由此看出,这种窗的性能可与哈明窗相比。另外,凯泽窗提供了可变的过渡带宽。由于贝塞尔函数的复杂性,这种窗的设计公式很难导出。幸运的是,凯泽开发了经验公式,下面我们将不加证明地给出。

设计公式

给定 $\omega_p\omega_s$,R_p 和 A_s

$$\text{归一化过渡带宽}=\Delta f\triangleq\frac{\omega_s-\omega_p}{2\pi}$$

$$\text{滤波器阶数 } M\simeq\frac{A_s-7.95}{14.36\Delta f}+1 \tag{7.30}$$

$$\text{参数 }\beta=\begin{cases}0.1102(A_s-8.7), & A_s\geqslant 50\\ 0.5842(A_s-21)^{0.4} & \\ \quad +0.07886(A_s-21), & 21<A_s<50\end{cases}$$

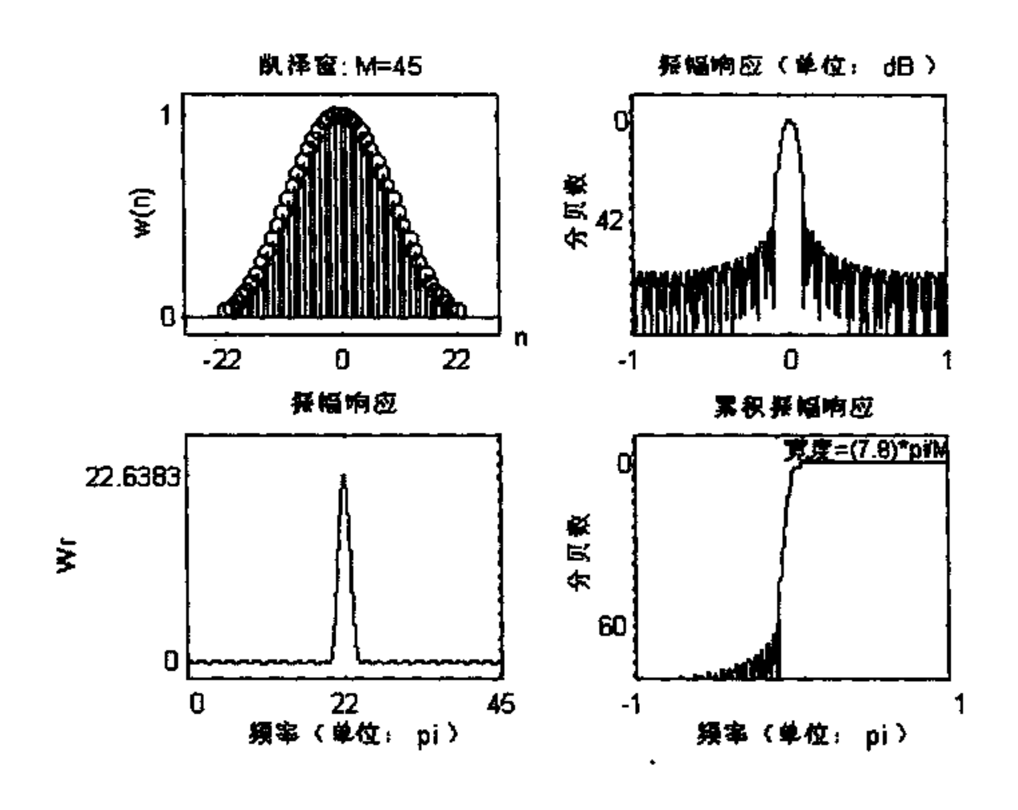


图 7.19　凯泽窗：$M=45,\beta=5.658$

MATLAB 实现

MATLAB 提供了几个子程序来实现本节中的窗函数。下面给出这些函数的简要说明。

- w = boxcar(M)　　数组 w 中返回 M 点矩形窗函数
- w = triang(M)　　数组 w 中返回 M 点 Bartlett(三角)窗函数
- w = hanning(M)　　数组 w 中返回 M 点汉宁窗函数
- w = hamming(M)　　数组 w 中返回 M 点哈明窗函数
- w = blackman(M)　　数组 w 中返回 M 点布莱克曼函数
- w = kaiser(M,beta)　　数组 w 中返回 beta 值 M 点凯泽窗函数

有了这些子程序，以窗技术为基础，我们就可利用 MATLAB 设计 FIR 滤波器了。同时，它还需要一个理想低通脉冲响应 $h_d(n)$。可以方便地用下面的简单子程序产生 $h_d(n)$。

```
function hd = ideal_lp(wc,M);
% 理想低通滤波器计算
% --------------------------------
% [hd] = ideal_lp(wc,M)
%   hd = 0 to M-1 之间的理想脉冲响应
%   wc = 截止频率(弧度)
%    M = 理想滤波器的长度
%
alpha = (M-1)/2;
```

```
n = [0:1:(M-1)];
m = n - alpha + eps;      % 加一个小数以避免零作除数
hd = sin(wc * m) ./ (pi * m);
```

在信号处理工具箱中，MATLAB 提供了一个子程序叫做 firl，它根据窗函数设计 FIR 滤波器。但是，这个子程序在学生版本中是没有的。为了显示数字滤波器频域中图形，MATLAB 提供了一个 freqz 子程序，这在前几章中已用过。利用这个子程序，我们开发一个修正版本，叫做 freqz_m，它给出绝对的和相对的 dB 值幅度响应，相位响应，以及群延迟响应。在下章中，我们将用到群延迟响应。

```
function [db,mag,pha,grd,w] = freqz_m(b,a);
% freqz 子程序的改进版本
% ----------------------------
% [db,mag,pha,grd,w] = freqz_m(b,a);
%   db = [0 到 pi 弧度]区间内的相对振幅(db)
%   mag = [0 到 pi 弧度]区间内的绝对振幅
%   pha = [0 到 pi 弧度]区间内的相位响应
%   grd = [0 到 pi 弧度]区间内的群延迟
%     w = [0 到 pi 弧度]区间内的 501 个频率样本向量
%     b = Ha(z)的分子多项式系数(对 FIR:b=h)
%     a = Ha(z)的分母多项式系数(对 FIR: a=[1])
%
[H,w] = freqz(b,a,1000,'whole');
    H = (H(1:1:501))'; w = (w(1:1:501))';
  mag = abs(H);
   db = 20 * log10((mag + eps)/max(mag));
  pha = angle(H);
  grd = grpdelay(b,a,w);
```

设计举例

现在，给出利用窗技术和 MATLAB 子程序进行滤波器设计的几个实例。

例 7.8 根据下列技术指标，设计一个数字 FIR 低通滤波器：

$$\omega_p = 0.2\pi, \qquad R_p = 0.25\text{dB}$$
$$\omega_s = 0.3\pi, \qquad A_s = 50\ \text{dB}$$

从表 7.1 中选出一个适当的窗函数，确定脉冲响应，并给出所设计的滤波器的频率响应图。

解： 哈明窗和布莱克曼窗均可提供大于 50dB 的衰减。我们选择哈明窗，它提供了较小的过渡带，因此，具有较小的阶数。尽管在设计中，没有使用通带波动值 $R_p = 0.25$dB，但必须检查设计的实际波动，验证它是否确实在给定容限内。下面的 MATLAB 程序给出设计步骤。

```
>> wp = 0.2 * pi; ws = 0.3 * pi;
>> tr_width = ws - wp
>> M = ceil(6.6 * pi/tr_width) + 1
M = 67
>> n=[0:1:M-1];
>> wc = (ws+wp)/2,%理想 LPF 截止频率
>> hd = ideal_lp(wc,M);
>> w_ham = (hamming(M))';
>> h = hd .* w_ham;
>> [db,mag,pha,grd,w] = freqz_m(h,[1]);
>> delta_w = 2 * pi/1000;
>> Rp = -(min(db(1:1:wp/delta_w+1)))          % 实际通带波动
Rp = 0.0394
>> As = -round(max(db(ws/delta_w+1:1:501)))    % 最小阻带衰减
As = 52
% 画图
>> subplot(1,1,1)
>> subplot(2,2,1); stem(n,hd); title('理想脉冲响应')
>> axis([0 M-1 -0.1 0.3]); xlabel('n');ylabel('hd(n)');
>> subplot(2,2,2); stem(n,w_ham);title('哈明窗')
>> axis([0 M-1 0 1.1]); xlabel('n');ylabel('w(n)');
>> subplot(2,2,3); stem(n,h);title('实际脉冲响应')
>> axis([0 M-1 -0.1 0.3]); xlabel('n'); ylabel('h(n)')
>> subplot(2,2,4); plot(w/pi,db);title('幅度响应(单位:dB)');grid
>> axis([0 1 -100 10]); xlabel('频率(单位:pi)'); ylabel('分贝数')
```

注意滤波器的长度 $M=67$,实际的阻带衰减为 52dB,还应当进行通带波动的验证。实际的通带波动为 0.0394dB。显然它是满足要求的。时域和频域的曲线如 7.20 所示。

例 7.9 利用例 7.8 给出的设计技术指标,选择凯泽窗,设计出所需的低通滤波器。

解: 设计步骤由下面的 MATLAB 程序给出。

```
>> wp = 0.2 * pi; ws = 0.3 * pi; As = 50;
>> tr_width = ws - wp;
>> M = ceil((As-7.95)/(14.36 * tr_width/(2 * pi))+1) + 1
M = 61
>> n=[0:1:M-1];
>> beta = 0.1102 * (As-8.7)
beta = 4.5513
>> wc = (ws+wp)/2;
>> hd = ideal_lp(wc,M);
```

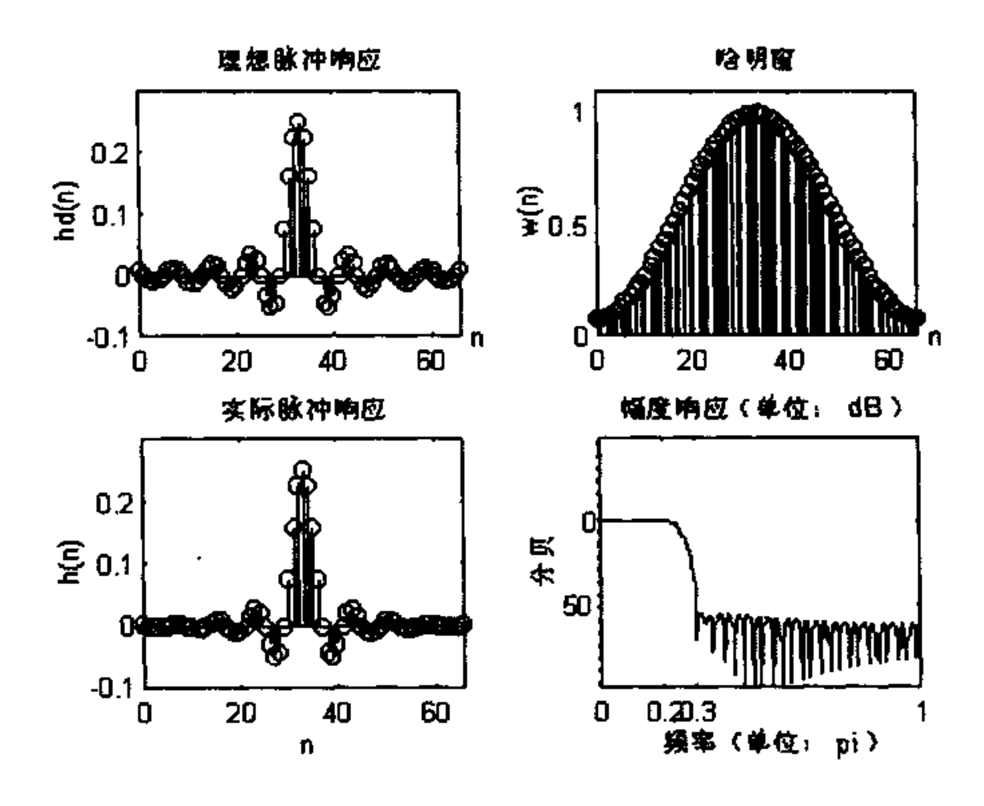


图 7.20 例 7.8 中的低通滤波器图

```
>> w_kai = (kaiser(M,beta))';
>> h = hd .* w_kai;
>> [db,mag,pha,grd,w] = freqz_m(h,[1]);
>> delta_w = 2*pi/1000;
>>As = -round(max(db(ws/delta_w+1:1:501))) % 最小阻带衰减
As = 52
% 画图
>>subplot(1,1,1)
>>subplot(2,2,1); stem(n,hd); title('理想脉冲响应')
>>axis([0 M-1 -0.1 0.3]); xlabel('n');ylabel('hd(n)');
>>subplot(2,2,2); stem(n,w_kai);title('凯泽窗')
>>axis([0 M-1 0 1.1]);xlabel('n'); ylabel('w(n)');
>>subplot(2,2,3); stem(n,h);title('实际脉冲响应')
>>axis([0 M-1 -0.1 0.3]); xlabel('n'); ylabel('h(n)')
>>subplot(2,2,4);plot(w/pi,db);title('幅度响应(单位: dB)');grid
>>axis([0 1 -100 10]); xlabel('频率(单位:pi)'); ylabel('分贝数')
```

凯泽窗的参数为 $M=61,\beta=4.5513$,实际的阻带衰减为 52dB。时域和频域的曲线在图 7.21 中给出。

例 7.10 设计下面的数字带通滤波器。

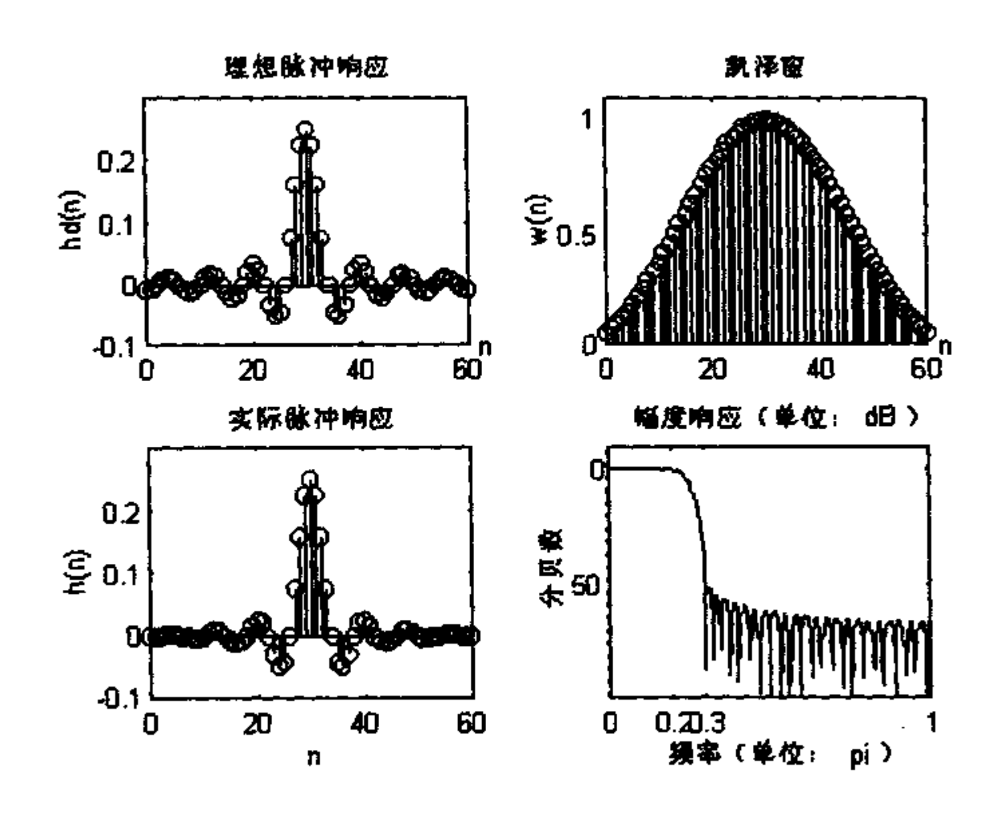


图 7.21 例 7.9 中的低通滤波器图

低端阻带边缘：　$\omega_{1s}=0.2\pi$，$A_s=60$ dB

低端通带边缘：　$\omega_{1p}=0.35\pi$，$R_p=1$ dB

高端通带边缘：　$\omega_{2p}=0.65\pi$，$R_p=1$ dB

高端阻带边缘：　$\omega_{2s}=0.8\pi$，$A_s=60$ dB

这些参数如图 7.22 所示。

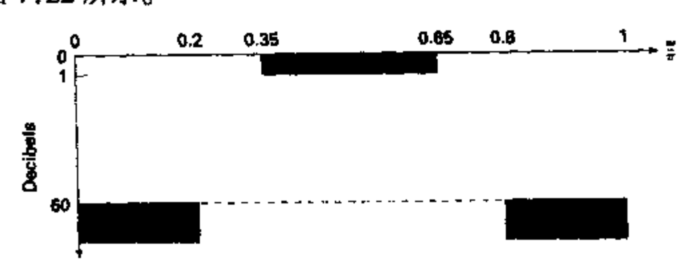


图 7.22 例 7.10 中的带通滤波器技术指标

解：存在两个过渡带，即 $\Delta\omega_1 \triangleq \Delta\omega_{1p}-\Delta\omega_{1s}$，和 $\Delta\omega_2 \triangleq \Delta\omega_{2s}-\Delta\omega_{2p}$，在窗设计中，这两个带宽必须相同。也就是说，对 $\Delta\omega_1$ 和 $\Delta\omega_2$ 不进行独立控制，因此 $\Delta\omega_1=\Delta\omega_2=\Delta\omega$。在此设计中，既可以用凯泽窗，也可以用布莱克曼窗。这里选用布莱克曼窗。同时也需要理想带通滤波器响应 $h_d(n)$。此脉冲响应可以由两个理想低通幅度响应得到，假如它们的相位响应相同，如图 7.23 所示。MATLAB 子程序 ideal_lp(wc,M)能有效地求出理想带通滤波器的脉冲响应。下面的 MATLAB 程序给出设计步骤。

```
>>ws1 = 0.2*pi; wp1 = 0.35*pi;
>>wp2 = 0.65*pi; ws2 = 0.8*pi;
>>As = 60;
>>tr_width = min((wp1-ws1),(ws2-wp2))
>>M = ceil(11*pi/tr_width) + 1
M = 75
>>n=[0:1:M-1];
>>wc1 = (ws1+wp1)/2; wc2 = (wp2+ws2)/2;
>>hd = ideal_lp(wc2,M) - ideal_lp(wc1,M);
>>w_bla = (blackman(M))';
>>h = hd .* w_bla;
>>[db,mag,pha,grd,w] = freqz_m(h,[1]);
>>delta_w = 2*pi/1000;
>>Rp = -min(db(wp1/delta_w+1:1:wp2/delta_w)) % 实际的通带波动
Rp = 0.0030
>>As = -round(max(db(ws2/delta_w+1:1:501))) % 最小阻带衰减
As = 75
% 画图
>>subplot(2,2,1); stem(n,hd); title('理想脉冲响应')
>>axis([0 M-1 -0.4 0.5]);xlabel('n');ylabel('hd(n)');
>>subplot(2,2,2); stem(n,w_bla);title('布莱克曼窗')
>>axis([0 M-1 0 1.1]);xlabel('n'); ylabel('w(n)');
>>subplot(2,2,3); stem(n,h);title('实际脉冲响应')
>>axis([0 M-1 -0.4 0.5]); xlabel('n'); ylabel('h(n)')
>>subplot(2,2,4);plot(w/pi,db);axis([0 1 -150 10]);
>>title('幅度响应(单位:dB)');grid;
>>xlabel('频率(单位: pi)'); ylabel('分贝数')
```

图 7.23 由两个低通滤波器得到的理想带通滤波器

布莱克曼窗的参数为 $M=61$,实际的阻带衰减为 75dB。时域和频域的曲线在图 7.24 中给出。

例 7.11 理想带阻滤波器的频率响应为:

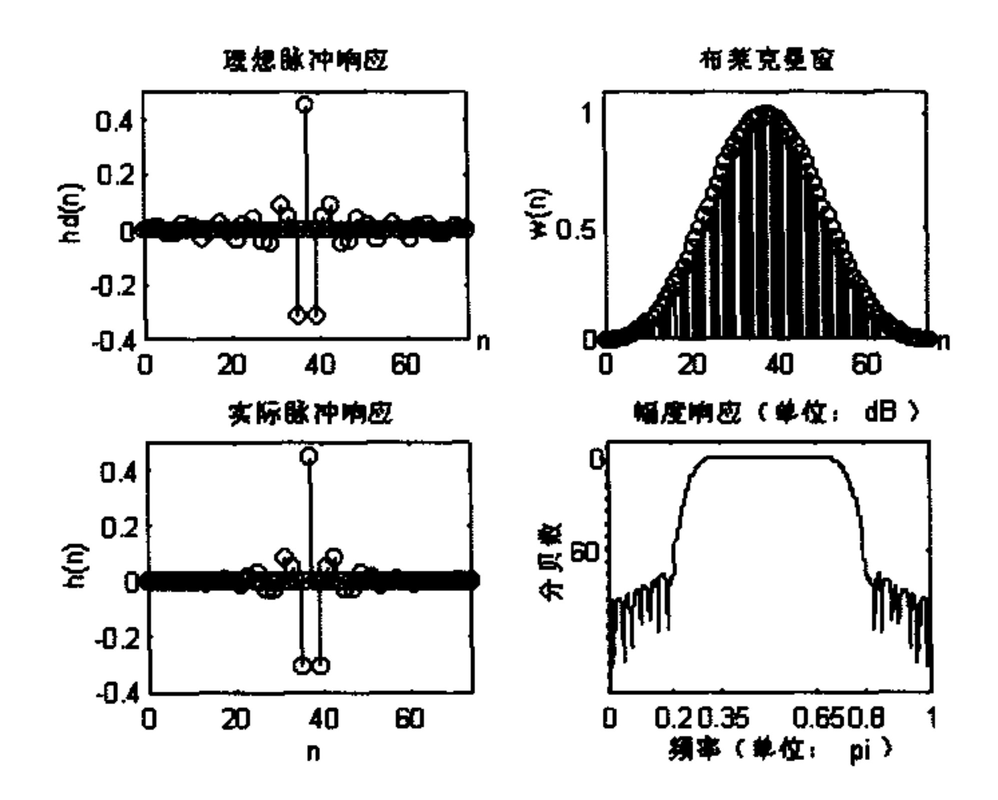


图 7.24 例 7.20 中的带通滤波器图

$$H_e(e^{j\omega}) = \begin{cases} 1, & 0 \leqslant |\omega| < \pi/3 \\ 0, & \pi/3 \leqslant |\omega| \leqslant 2\pi/3 \\ 1, & 2\pi/3 < |\omega| \leqslant \pi \end{cases}$$

用凯泽窗设计一个长度为 M 的带阻滤波器，它的阻带衰减为 60dB。

解：在以上技术指标中，没有给定过渡带宽。它将由凯泽窗的长度 $M = 45$ 和参数 β 确定。根据设计公式(7.30)，由 A_s 可确定 β，即：

$$\beta = 0.1102(A_s - 8.7)$$

采用与图 7.23 类似的方法，也可从理想低通响应确定理想带阻脉冲响应。下面进行凯泽窗设计，并检查是否得到最小阻带衰减。这由下面的 MATLAB 程序给出。

```
>>M = 45; As = 60; n=[0:1:M-1];
>>beta = 0.1102*(As-8.7)
beta = 5.6533
>>w_kai = (kaiser(M,beta))';
>>wc1 = pi/3; wc2 = 2*pi/3;
>>hd = ideal_lp(wc1,M) + ideal_lp(pi,M) - ideal_lp(wc2,M);
>>h = hd .* w_kai;
>>[db,mag,pha,grd,w] = freqz_m(h,[1]);
```

```
> > subplot(1,1,1);
> > subplot(2,2,1);stem(n,hd);title('理想脉冲响应')
> > axis([-1,M-0.2 0.8]);xlabel('n');ylabel(,hd(n)')
> > subplot(2,2,2);stem(n,W-Kai);title('凯泽窗')
> > axis([-1,M0 1.1]);xlabel('n');ylabel(,w(n)')
> > subplot(2,2,3);stem(n,h);title('实际脉冲响应')
> > axis([-1,M  -0.2 0.8]);xlabel('n');ylabel('h(n)')
> > subplot(2,2,4);plot(w/pi,db);axis([0 1  -80 10]);
> > title('幅度响应(单位:dB)');grid;
> > xlabel('频率(单位: pi)'); ylabel('分贝数')
```

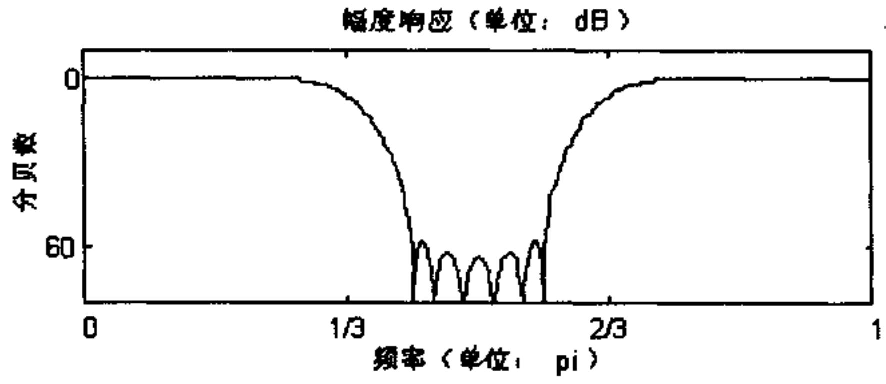


图 7.25　例 7.11 中的带阻滤波器幅度响应　$\beta=5.6533$

参数 β 等于 5.6533。从图 7.25 的幅度曲线可知最小阻带衰减小于 60dB。因此,需要增加 β,使得衰减增至 60dB。所需要的值为 $\beta=5.9533$。

```
> > M = 45; As = 60; n=[0:1:M-1];
> > beta = 0.1102*(As-8.7)+0.3
beta = 5.9533
> > w_kai = (kaiser(M,beta))';
> > wc1 = pi/3; wc2 = 2*pi/3;
> > hd = ideal_lp(wc1,M) + ideal_lp(pi,M) - ideal_lp(wc2,M);
> > h = hd .* w_kai;
> > [db,mag,pha,grd,w] = freqz_m(h,[1]);
> > subplot(1,1,1);
> > subplot(2,2,1); stem(n,hd); title('理想脉冲响应')
> > axis([-1 M  -0.2 0.8]);xlabel('n'); ylabel('hd(n)');
> > subplot(2,2,2); stem(n,w_kai);title('凯泽窗')
> > axis([-1 M 0 1.1]);xlabel('n');ylabel('w(n)');
> > subplot(2,2,3); stem(n,h);title('实际脉冲响应')
> > axis([-1 M  -0.2 0.8]); xlabel('n'); ylabel('h(n)')
> > subplot(2,2,4);plot(w/pi,db);axis([0 1  -80 10]);
> > title('幅度响应(单位:db)');grid;
```

```
> > xlabel('频率(单位: pi )'); ylabel('分贝数')
```

图 7.26 给出时域和频域的各图。可以看出,所设计的滤波器满足要求。

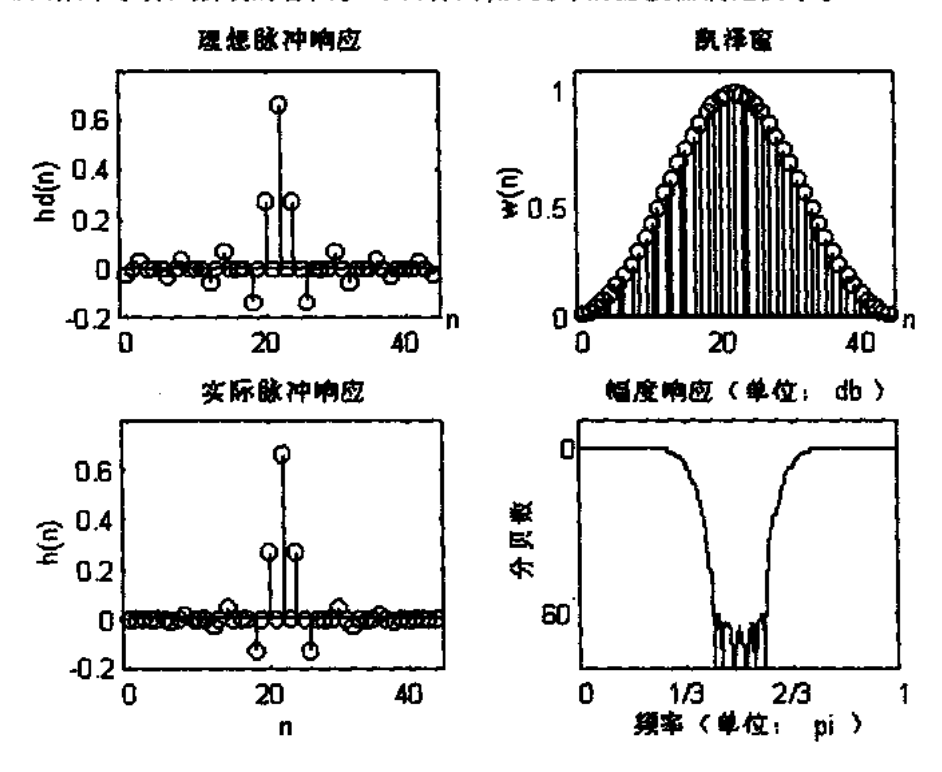


图 7.26 例 7.11 中的带阻滤波器: $\beta = 5.9533$

例 7.12 下式给出理想数字微分器的频率响应:

$$H_d(e^{j\omega}) = \begin{cases} j\omega, & 0 < \omega \leqslant \pi \\ -j\omega, & -\pi < \omega < 0 \end{cases} \tag{7.31}$$

用长度为 21 的哈明窗,设计一个数字 FIR 微分器。画出时域和频域响应图。

解:具有线性相位的数字微分器的理想脉冲响应为:

$$\begin{aligned} h_d(n) &= F[H_d(e^{j\omega})e^{j\alpha\omega}] = \frac{1}{2\pi}\int_{-\pi}^{\pi} H_d(e^{j\omega})e^{-j\alpha\omega}e^{j\omega n}d\omega \\ &= \frac{1}{2\pi}\int_{-\pi}^{0}(-j\omega)e^{-j\alpha\omega}e^{j\omega n}d\omega + \frac{1}{2\pi}\int_{0}^{\pi}(j\omega)e^{-j\alpha\omega}e^{j\omega n}d\omega \\ &= \begin{cases} \dfrac{\cos\pi(n-\alpha)}{(n-\alpha)}, & n \neq \alpha \\ 0, & n = \alpha \end{cases} \end{aligned}$$

用 MATLAB 可以实现上面的脉冲响应,结合哈明窗可设计要求的微分器。如果 M 为偶数,$\alpha = (M-1)/2$ 就不是一个整数,则对所有的 n,$h_d(n)$ 均为零。因此,M 必须是奇数。这

是一个 3 型的线性相位 FIR 滤波器。由于 3 型滤波器的 $H_r(\pi)=0$,此滤波器不是一个全通带的微分器。

```
>>M = 21; alpha = (M-1)/2;
>>n = 0:M-1;
>>hd = (cos(pi*(n-alpha)))./(n-alpha); hd(alpha+1)=0;
>>w_ham = (hamming(M))';
>>h = hd .* w_ham;
>>[Hr,w,P,L] = Hr_Type3(h);
% 画图
>>subplot(1,1,1);
>>subplot(2,2,1); stem(n,hd); title('理想脉冲响应')
>>axis([-1 M -1.2 1.2]);xlabel('n'); ylabel('hd(n)');
>>subplot(2,2,2); stem(n,w_ham);title('哈明窗')
>>axis([-1 M 0 1.2]); xlabel('n');ylabel('w(n)');
>>subplot(2,2,3); stem(n,h);title('实际脉冲响应')
>>axis([-1 M -1.2 1.2]); xlabel('n'); ylabel('h(n)')
>>subplot(2,2,4);plot(w/pi,Hr/pi); title('振幅响应');grid;
>>xlabel('频率(单位:pi)'); ylabel('斜率(单位:pi)')
>>axis([0 1 0 1]);
```

各曲线在图 7.27 中给出。

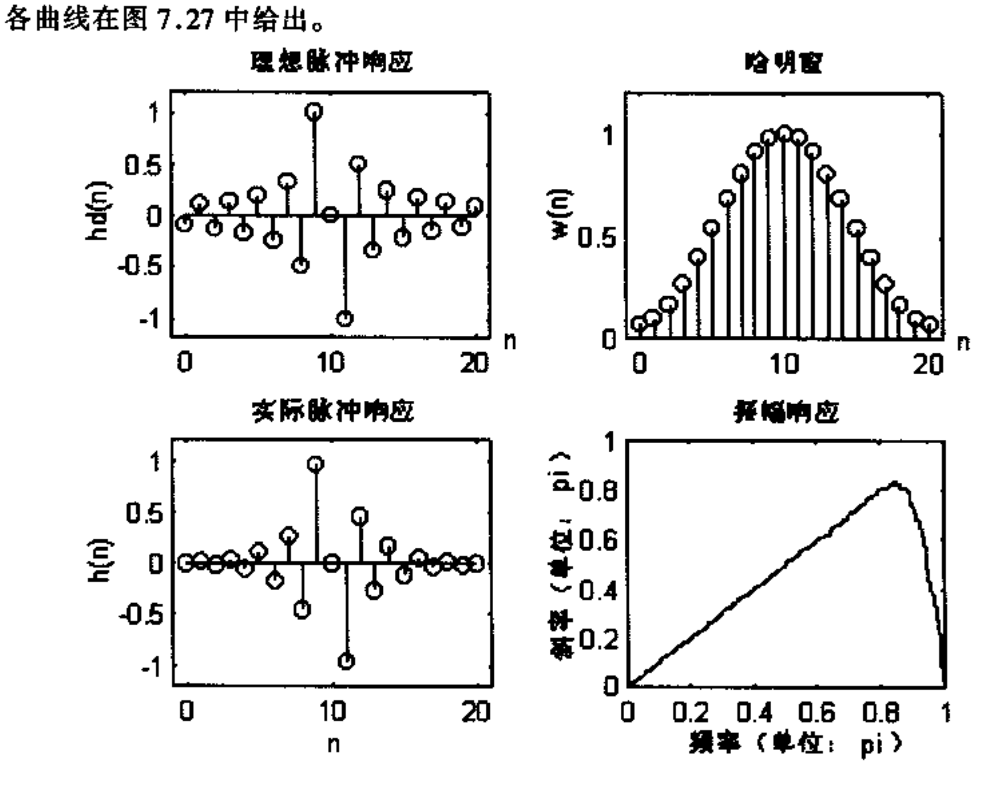


图 7.27　例 7.12 中的 FIR 微分器设计

例 7.13 用汉宁窗设计一个长度为25的数字希尔伯特变换器。

解:线性相位希尔伯特变换器的理想频率响应为:

$$H_d(e^{j\omega}) = \begin{cases} -je^{-j\alpha\omega}, & 0 < \omega < \pi \\ +je^{-j\alpha\omega}, & -\pi < \omega < 0 \end{cases} \tag{7.32}$$

经过逆变换后,理想脉冲响应为:

$$h_d(n) = \begin{cases} \dfrac{2}{\pi} \dfrac{\sin^2 \pi(n-\alpha)/2}{n-\alpha}, & n \neq \alpha \\ 0, & n = \alpha \end{cases}$$

这可由 MATLAB 很容易地实现。由于 $M = 25$,所设计的滤波器是 3 型的。

```
>>M = 25; alpha = (M-1)/2;
>>n = 0:M-1;
>>hd = (2/pi)*((sin((pi/2)*(n-alpha)).^2)./(n-alpha)); hd(alpha+1)=0;
>>w_han = (hanning(M))';
>>h = hd .* w_han;
>>[Hr,w,P,L] = Hr_Type3(h);
% 画图
>>subplot(1,1,1);
>>subplot(2,2,1); stem(n,hd); title('理想脉冲响应')
>>axis([-1 M -1.2 1.2]);xlabel('n'); ylabel('hd(n)');
>>subplot(2,2,2); stem(n,w_han);title('汉宁窗')
>>axis([-1 M 0 1.2]);xlabel('n'); ylabel('w(n)');
>>subplot(2,2,3); stem(n,h);title('实际脉冲响应')
>>axis([-1 M -1.2 1.2]);xlabel('n'); ylabel('h(n)')
>>w = w'; Hr = Hr';
>>w = [-fliplr(w), w(2:501)]; Hr = [-fliplr(Hr), Hr(2:501)];
>>subplot(2,2,4);plot(w/pi,Hr); title('振幅响应');grid;
>>xlabel('频率(单位:pi)'); ylabel('Hr')
>>axis([-1 1 -1.1 1.1]);
```

滤波器响应的曲线如图 7.28 所示。振幅响应是在区间 $-\pi \leqslant \omega \leqslant \pi$ 给出的。

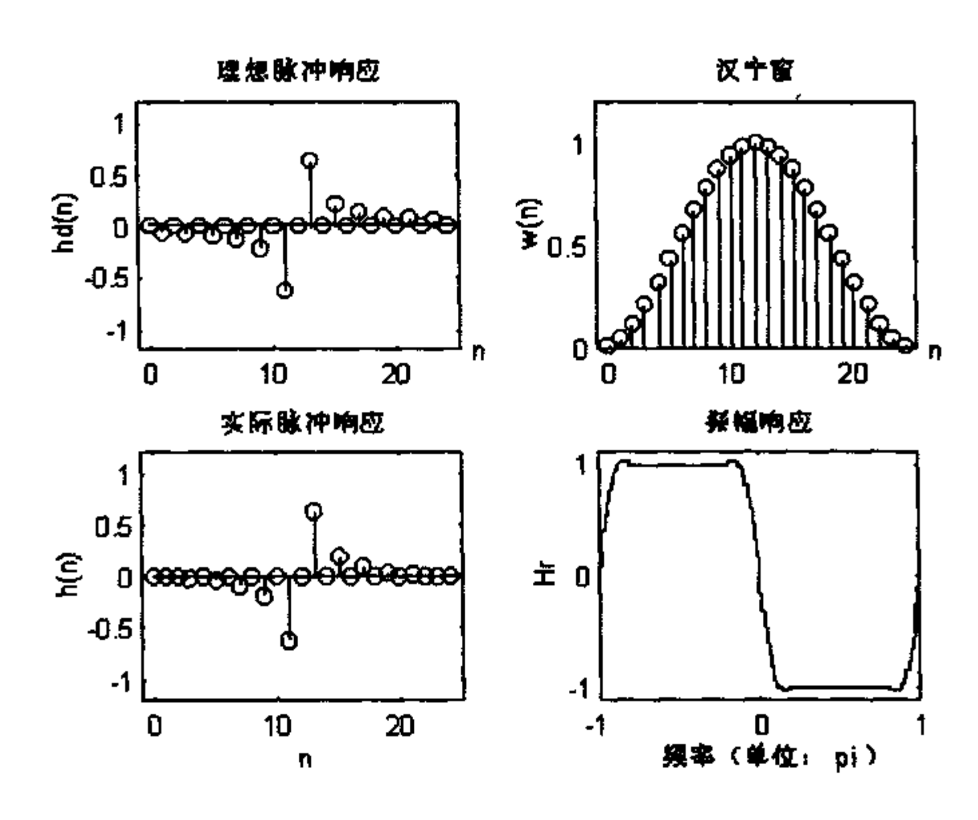


图 7.28 例 7.13 中的 FIR 希尔伯特变换器设计

频率采样设计技术

这种方法所依据的事实是，系统函数 $H(z)$ 可根据频率响应 $H(e^{j\omega})$ 的样本 $H(k)$ 得到。而且，此设计技术与第六章讨论的频率采样结构相吻合。设 $h(n)$ 为 M 点滤波器的脉冲响应，$H(k)$ 是 M 点 DFT，$H(z)$ 是它的系统函数，则由(6.12)，得：

$$H(z) = \sum_{n=0}^{M-1} h(n) z^{-n} = \frac{1 - z^{-M}}{M} \sum_{k=0}^{M-1} \frac{H(k)}{1 - z^{-1} e^{j2\pi k/M}} \tag{7.33}$$

并且

$$H(e^{j\omega}) = \frac{1 - e^{-j\omega M}}{M} \sum_{k=0}^{M-1} \frac{H(k)}{1 - e^{-j\omega} e^{j2\pi k/M}} \tag{7.34}$$

其中

$$H(k) = H(e^{j2\pi k/M}) = \begin{cases} H(0), & k = 0 \\ H^*(M - k), & k = 1, \cdots, M - 1 \end{cases}$$

对于线性相位滤波器，有

$$h(n) = \pm h(M-1-n), n = 0,1,\cdots,M-1$$

其中,正号对应于1型和2型,负号对应于3型和4型滤波器。因此,$H(k)$变为:

$$H(k) = H_r\left(\frac{2\pi k}{M}\right)e^{j\angle H(k)} \tag{7.35}$$

其中

$$H_r\left(\frac{2\pi k}{M}\right) = \begin{cases} H_r(0), & k = 0 \\ H_r\left(\frac{2\pi(M-k)}{M}\right), & k = 1,\cdots,M-1 \end{cases} \tag{7.36}$$

并且

$$\angle H(k) = \begin{cases} -\left(\frac{M-1}{2}\right)\left(\frac{2\pi k}{M}\right), & k = 0,\cdots,\left\lfloor \frac{M-1}{2} \right\rfloor \\ +\left(\frac{M-1}{2}\right)\frac{2\pi}{M}(M-k), & k = \left\lfloor \frac{M-1}{2} \right\rfloor + 1,\cdots,M-1 \end{cases} \quad (1\text{型和}2\text{型}) \tag{7.37}$$

或

$$\angle H(k) = \begin{cases} \left(\pm\frac{\pi}{2}\right) - \left(\frac{M-1}{2}\right)\left(\frac{2\pi k}{M}\right), & k = 0,\cdots,\left\lfloor \frac{M-1}{2} \right\rfloor \\ -\left(\pm\frac{\pi}{2}\right) + \left(\frac{M-1}{2}\right)\frac{2\pi}{M}(M-k), & \\ & k = \left\lfloor \frac{M-1}{2} \right\rfloor + 1,\cdots,M-1 \end{cases} \quad (3\text{型和}4\text{型}) \tag{7.38}$$

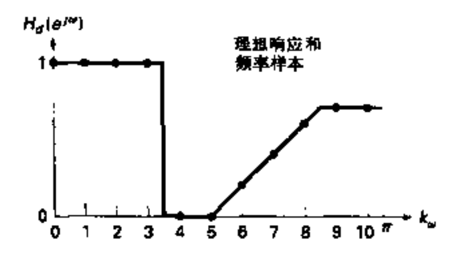
最后,有

$$h(n) = \text{IDFT}\{H(k)\} \tag{7.39}$$

在许多教材(例如,[19,20,16])中,明确给出了已知 $H(k)$计算 $h(n)$的公式。我们将根据(7.39),利用 MATLAB 的 ifft 函数计算 $h(n)$。

基本思想 给定理想低通滤波器 $H_d(e^{j\omega})$,先选择滤波器长度 M,然后对 $H_d(e^{j\omega})$在0到 2π 上的 M 个等间隔频率上采样,根据(7.34),通过对样本 $H(k)$的内插,得到实际响应 $H(e^{j\omega})$。如图7.29所示。脉冲响应由(7.39)得到。对其他的选频滤波器可采用类似的步骤。这种思想还可推广到逼近任意频域技术指标。

由图7.29,可知



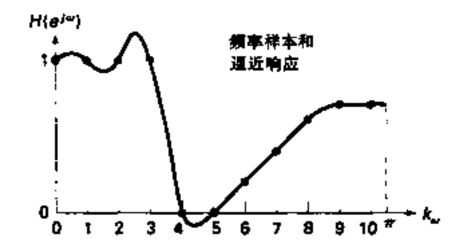


图 7.29 频域采样技术图解

1. 在采样频率上的逼近误差为零,也就是理想的和实际响应的差为零。

2. 其余频率上的逼近误差取决于理想响应的形状:理想响应的轮廓越陡,则逼近误差越大。

3. 靠近带的边缘的误差越大,在带内的误差越小。

目前有两种设计方法,第一种直接用上面的基本思想,而对逼近误差不加任何限制;也就是说,无论设计所得的误差为多大,我们都接受,这种方法叫做朴素设计法。第二种方法则通过改变过渡带的样本值,努力使阻带中的误差极小化,这就产生了一个较好的设计,叫做最优设计方法。

朴素设计法

在这种方法中,使 $H(k)=H_d(e^{j2\pi k/M}),k=0,\ldots,M-1$,然后利用(7.35)至(7.39)得到脉冲响应 $h(n)$。

例 7.14 考虑例7.8中的低通滤波器技术指标:

$$\omega_p=0.2\pi,\qquad R_p=0.25\quad \text{dB}$$
$$\omega_s=0.3\pi,\qquad A_s=50\quad \text{dB}$$

用频率采样方法设计一个 FIR 滤波器。

解:选择 $M=20$,这样,在 ω_p 处有一个频率样本,也就是在 $k=2$ 时:

$$\omega_p=0.2\pi=\frac{2\pi}{20}2$$

下一个样本在 ω_s 处,即 $k=3$ 时:

$$\omega_s=0.3\pi=\frac{2\pi}{20}3$$

因此,在通带$[0\leqslant\omega\leqslant\omega_p]$上有三个样本,阻带$[\omega_s\leqslant\omega\leqslant\pi]$上有 7 个样本。由(7.36)有

$$H_r(k)=[1,1,1,\underbrace{0,\cdots,0}_{15\text{个零}},1,1]$$

由于 $M=20, \alpha=(20-1)/2=9.5$,因此这是一个 2 型线性相位滤波器,根据(7.37)有:

$$\angle H(k)=\begin{cases}-9.5\dfrac{2\pi}{20}k=-0.95\pi k, & 0\leqslant k\leqslant 9\\ +0.95\pi(20-k), & 10\leqslant k\leqslant 19\end{cases}$$

从(7.35)得到 $H(k)$,然后根据(7.39)确定脉冲响应 $h(n)$。下面给出 MATLAB 程序:

```
>>M = 20; alpha = (M-1)/2; l = 0:M-1; wl = (2*pi/M)*l;
>>Hrs = [1,1,1,zeros(1,15),1,1];%理想振幅响应采样
>>Hdr = [1,1,0,0]; wdl = [0,0.25,0.25,1];%理想振幅响应
>>k1 = 0:floor((M-1)/2); k2 = floor((M-1)/2)+1:M-1;
>>angH = [-alpha*(2*pi)/M*k1, alpha*(2*pi)/M*(M-k2)];
>>H = Hrs.*exp(j*angH);
>>h = real(ifft(H,M));
>>[db,mag,pha,grd,w] = freqz_m(h,1);
>>[Hr,ww,a,L] = Hr_Type2(h);
>>subplot(1,1,1)
>>subplot(2,2,1);plot(wl(1:11)/pi,Hrs(1:11),'o',wdl,Hdr);
>>axis([0,1,-0.1,1.1]); title('频率样本: M=20')
>>xlabel('频率(单位:pi)'); ylabel('Hr(k)')
>>subplot(2,2,2); stem(l,h); axis([-1,M,-0.1,0.3])
>>title('脉冲响应');xlabel('n');ylabel('h(n)');
>>subplot(2,2,3); plot(ww/pi,Hr,wl(1:11)/pi,Hrs(1:11),'o');
>>axis([0,1,-0.2,1.2]); title('振幅响应')
>>xlabel('频率(单位:pi)'); ylabel('Hr(w)')
>>subplot(2,2,4);plot(w/pi,db); axis([0,1,-60,10]); grid
>>title('幅度响应'); xlabel('频率(单位:pi)');ylabel('分贝数');
```

频域及其时域曲线如图 7.30 所示。注意到最小阻带衰减为 16dB,显然是不能接受的。如果增大 M,则在过渡带中会有样本,而它的频率响应是不能精确知道的。因此,在实际中很少用到朴素设计法。

最优设计法

为了得到更大的衰减,必须增大 M,并且使过渡带中的样本成为自由样本,也就是说,可以改变它们的样本值,以便在给定的 M 和过渡带宽下得到最大的衰减。这个问题称为优化问题,利用线性规划技术可以解决它。我们通过下面的例子来说明改变过渡带样本值对设计的影响。

例 7.15 用最优设计法,为例 7.14 设计一个较好的低通滤波器。

解:选择 $M=40$,这样在过渡带 $0.2\pi<\omega<0.3\pi$ 中有一个样本。因为 $\omega_1 \triangleq 2\pi/40$,过渡

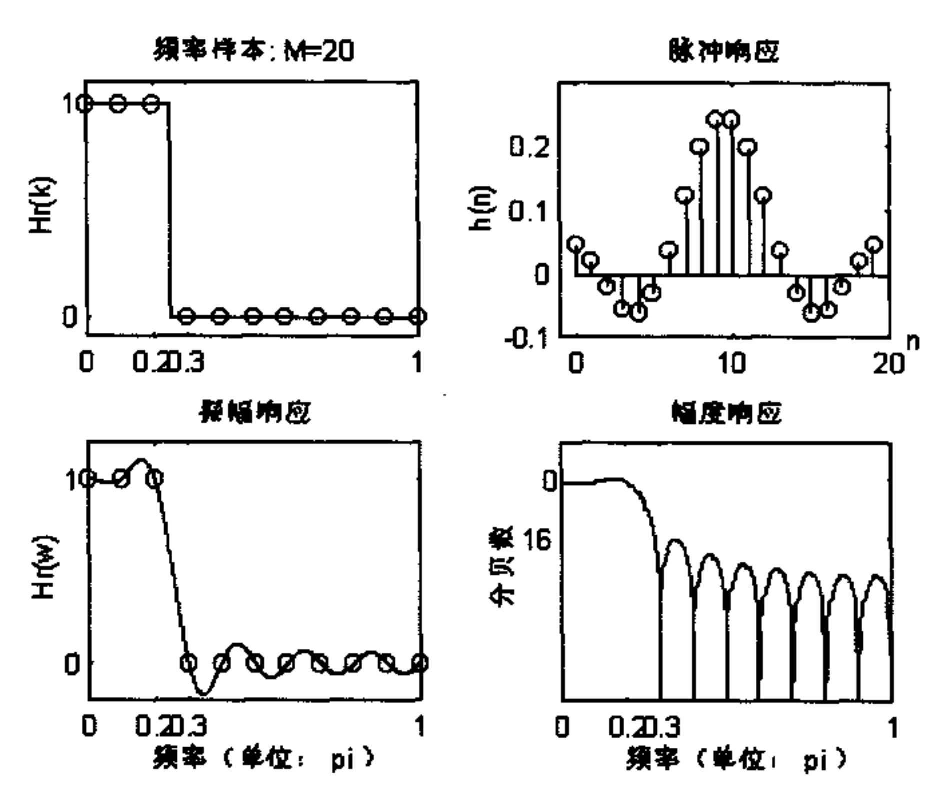


图 7.30 朴素频率采样设计法

带样本在 $k=5$ 及 $k=40-5=35$ 处。假设用 T_1 表示这些样本的值，$0<T_1<1$；被采样的振幅响应为：

$$H_r(k)=[1,1,1,1,1,T_1,\underbrace{0,\cdots,0}_{29个零},T_1,1,1,1,1]$$

由于 $\alpha=(40-1)/2=19.5$，相位响应的样本为：

$$\angle H(k)=\begin{cases}-19.5\dfrac{2\pi}{40}k=-0.975\pi k, & 0\leqslant k\leqslant 19\\ +0.975\pi(40-k), & 20\leqslant k\leqslant 39\end{cases}$$

改变 T_1，以得到最好的最小阻带衰减，这会导致过渡带宽的增大。首先看 $T_1=0.5$ 会产生什么结果。

```
% T1 = 0.5
>>M = 40; alpha = (M-1)/2;
>>Hrs = [ones(1,5),0.5,zeros(1,29),0.5,ones(1,4)];
>>Hdr = [1,1,0,0]; wdl = [0,0.25,0.25,1];
>>k1 = 0:floor((M-1)/2); k2 = floor((M-1)/2)+1:M-1;
>>angH = [-alpha*(2*pi)/M*k1, alpha*(2*pi)/M*(M-k2)];
>>H = Hrs.*exp(j*angH);
```

```
>>h = real(ifft(H,M));
```

从图 7.31 知，最小的阻带衰减为 30dB，它比朴素设计法所产生的衰减好一点，但还没有达到要求的 50dB。经过手动地变化 T_1 值，得到它的最佳值（尽管线性规划技术更有效，但这里没有使用它）。近似最优解在 $T_1 = 0.39$ 处。

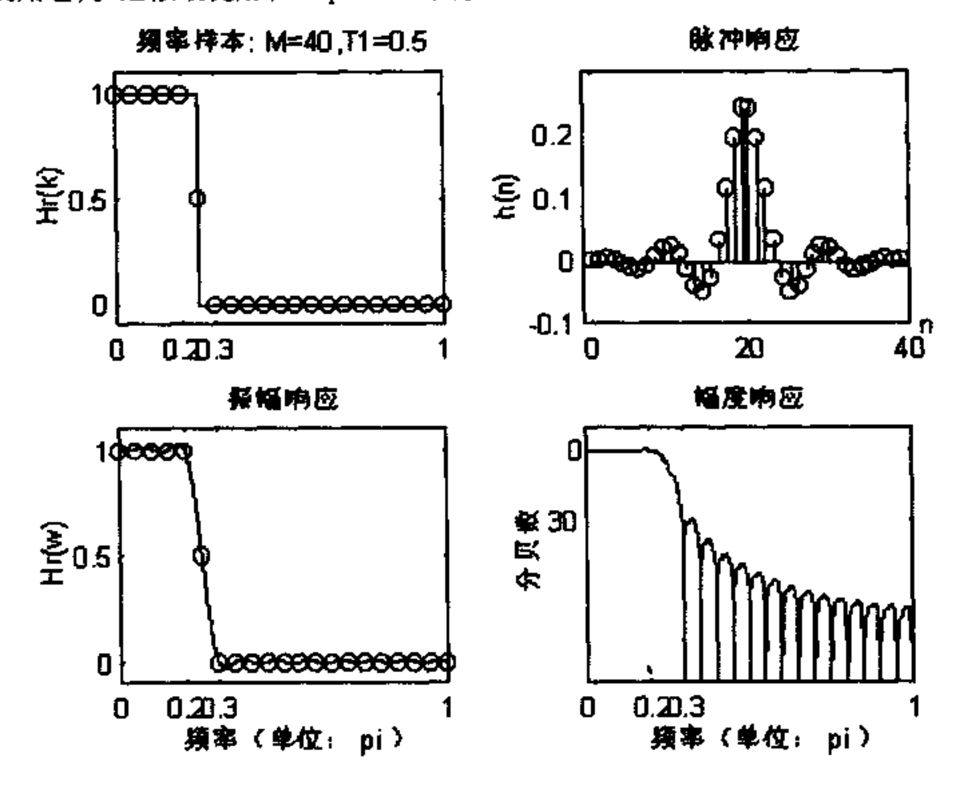


图 7.31　最优频率设计法：$T_1 = 0.5$

```
% T1 = 0.39
>>M = 40; alpha = (M-1)/2;
>>Hrs = [ones(1,5),0.39,zeros(1,29),0.39,ones(1,4)];
>>k1 = 0:floor((M-1)/2); k2 = floor((M-1)/2)+1:M-1;
>>angH = [-alpha*(2*pi)/M*k1, alpha*(2*pi)/M*(M-k2)];
>>H = Hrs.*exp(j*angH);
>>h = real(ifft(H,M));
```

从图 7.32 看出，优化的阻带衰减为 43dB，很明显，为了进一步增加衰减，在过渡带中，必须改变多于一个的样本值。

显然，这种方法的优越性在于，通过改变一个样本，可以得到一个较好的设计。实际中，一般来讲，过渡带宽很小，包含一到两个样本。因此最多需要优化两个样本，来得到最大的最小阻带衰减。这与在绝对意义下使最大旁瓣幅度最小是等价的。这个优化问题也叫做最大最小化问题。Rabiner 等[20]解决了这个问题。并以表格的形式给出了解。有的书也选择性地给出了一些表[参考文献 19，附录 B]。这个问题也可由 MATLAB 解决，但它需要用到优化工具

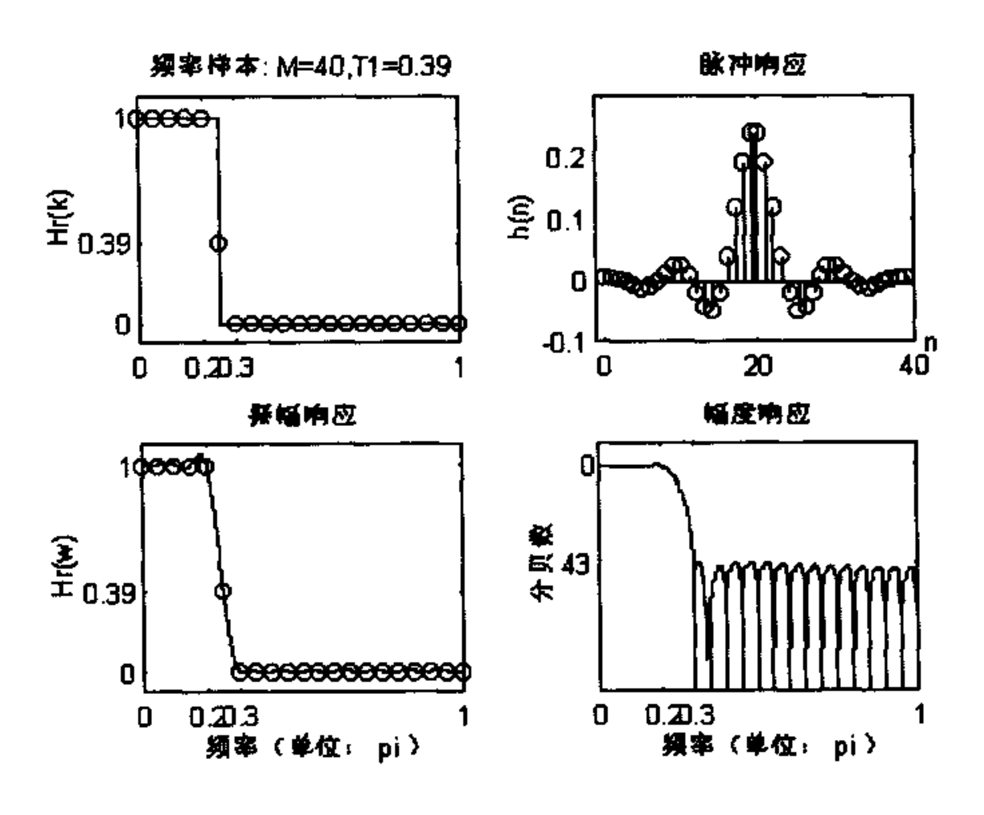


图 7.32 最优频率设计法: $T_1=0.39$

箱。在后一节中,将以更通用方式研究这个问题。下面的例子将说明这些表的用法。

例 7.16 再次考察例7.14中的低通滤波器设计,这里用过渡带中的两个样本来解决,以此得到较好的阻带衰减。

解:选择 $M=60$,这样在过渡带中有两个样本。设其值为 T_1 和 T_2,则 $H_r(\omega)$为:

$$H(\omega)=[\underbrace{1,\cdots,1}_{7个1},T_1,T_2,\underbrace{0,\cdots,0}_{43个0},T_2,T_1,\underbrace{1,\cdots,1}_{6个1}]$$

由表[19,附录 B]得,$T_1=0.5925$,$T_2=0.1099$,根据这些值,用 MATLAB 计算 $h(n)$:

```
>>M = 60; alpha = (M-1)/2; l = 0:M-1; wl = (2*pi/M)*l;
>>Hrs = [ones(1,7),0.5925,0.1099,zeros(1,43),0.1099,0.5925,ones(1,6)];
>>Hdr = [1,1,0,0]; wdl = [0,0.2,0.3,1];
>>k1 = 0:floor((M-1)/2); k2 = floor((M-1)/2)+1:M-1;
>>angH = [-alpha*(2*pi)/M*k1, alpha*(2*pi)/M*(M-k2)];
>>H = Hrs.*exp(j*angH);
>>h = real(ifft(H,M));
>>[db,mag,pha,grd,w] = freqz_m(h,1);
>>[Hr,ww,a,L] = Hr_Type2(h);
```

时域和频域曲线如图 7.33 所示。现在的阻带衰减为 63dB,这是可以接受的。

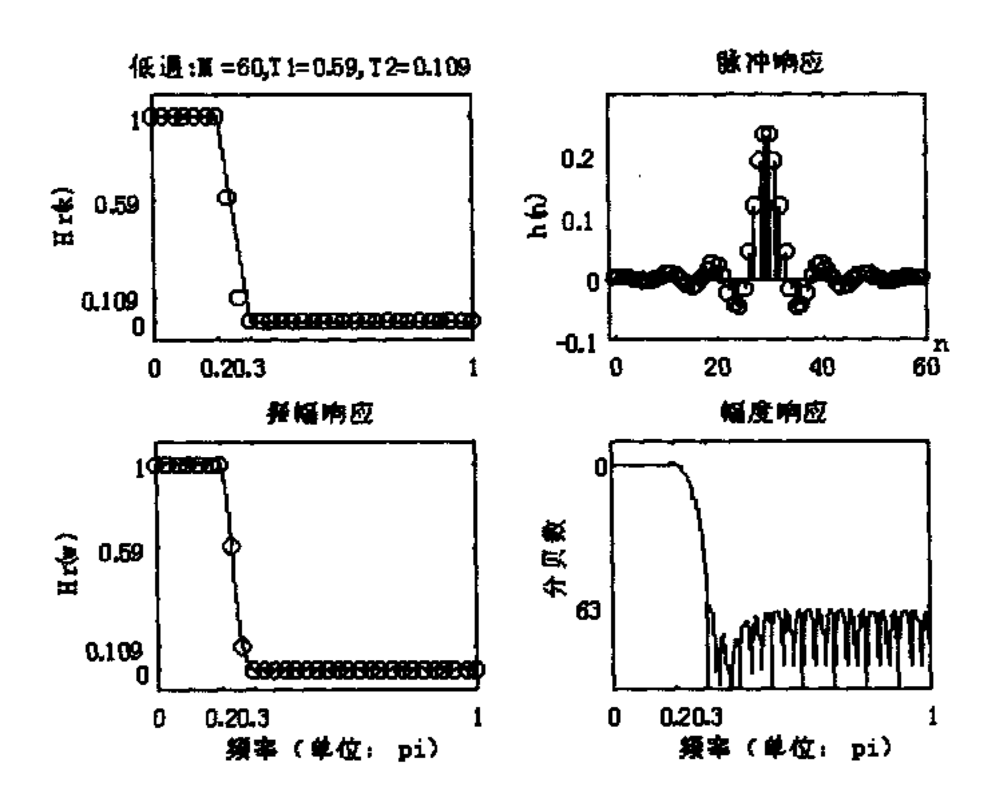


图 7.33　例 7.16 中的低通滤波器设计图

例 7.17　用频率采样技术设计例7.10中的带通滤波器，其技术指标为：

$$\text{低阻带边缘：}\omega_{1s} = 0.2\pi,\ A_s = 60\ \text{dB}$$
$$\text{低通带边缘：}\omega_{1p} = 0.35\pi,\ R_p = 1\ \text{dB}$$
$$\text{高通带边缘：}\omega_{2p} = 0.65\pi,\ R_p = 1\ \text{dB}$$
$$\text{高阻带边缘：}\omega_{2s} = 0.8\pi,\ A_s = 60\ \text{dB}$$

解：选择 $M=40$，这样在过渡带中有两个样本。设低过渡带中的频率样本值为 T_1 和 T_2，则幅度响应中的各样本为：

$$H_r(\omega) = [\underbrace{0,\cdots,0}_{5}, T_1, T_2, \underbrace{1,\cdots,1}_{7}, T_2, T_1, \underbrace{0,\cdots,0}_{9}, T_1, T_2, \underbrace{1,\cdots,1}_{7}, T_2, T_1, \underbrace{0,\cdots,0}_{4}]$$

$M=40$ 而且通带中有 7 个样本的 T_1 和 T_2 的最优值为[19,附录 B]：

$$T_1 = 0.109021,\ T_2 = 0.59417456$$

其 MATLAB 程序为：

```
>>M = 40; alpha = (M-1)/2; l = 0:M-1; wl = (2*pi/M)*l;
>>T1 = 0.109021; T2 = 0.59417456;
>>Hrs=[zeros(1,5),T1,T2,ones(1,7),T2,T1,zeros(1,9),T1,T2,ones(1,7),T2,T1,
```

```
zeros(1,4)];
    >>Hdr = [0,0,1,1,0,0]; wdl = [0,0.2,0.35,0.65,0.8,1];
    >>k1 = 0:floor((M-1)/2); k2 = floor((M-1)/2)+1:M-1;
    >>angH = [-alpha*(2*pi)/M*k1, alpha*(2*pi)/M*(M-k2)];
    >>H = Hrs.*exp(j*angH);
    >>h = real(ifft(H,M));
    >>[db,mag,pha,grd,w] = freqz_m(h,1);
    >>[Hr,ww,a,L] = Hr_Type2(h);
```

图 7.34 表明这是一个可以接受的带通滤波器设计。

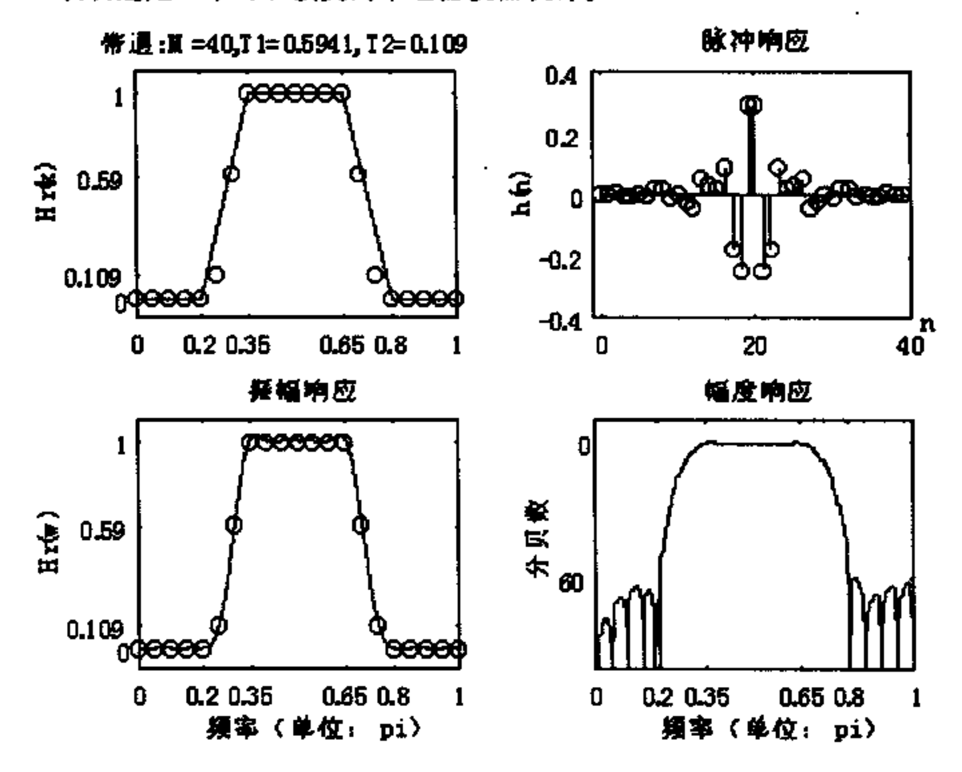


图 7.34 例 7.17 中的带通滤波器设计曲线

例 7.18 设计下面的高通滤波器：

$$\text{阻带边缘：}\omega_s = 0.6\pi,\ A_s = 50\ \text{dB}$$

$$\text{通带边缘：}\omega_p = 0.8\pi,\ R_p = 1\ \text{dB}$$

解：回想对于高通滤波器，M 必须为奇数(1 型滤波器)。为了在过渡带得到两个样本，选择 $M=33$。这样的选择不可能在 ω_s 及 ω_p 处得到频率样本。幅度响应的样本为：

$$H_r(k) = [\underbrace{0,\cdots,0}_{11}, T_1, T_2, \underbrace{1,\cdots,1}_{8}, T_2, T_1, \underbrace{0,\cdots,0}_{10}]$$

同时，相位样本为：

$$\angle H(k)=\begin{cases}-\dfrac{33-1}{2}\dfrac{2\pi}{33}k=-\dfrac{32}{33}\pi k, & 0\leqslant k\leqslant 16\\ +\dfrac{32}{33}\pi(33-k), & 17\leqslant k\leqslant 32\end{cases}$$

过渡带样本的最优值为 $T_1=0.1095$，$T_2=0.598$，根据这些值，MATLAB 设计为：

```
>>M = 33; alpha = (M-1)/2; l = 0:M-1; wl = (2*pi/M)*l;
>>T1 = 0.1095; T2 = 0.598;
>>Hrs = [zeros(1,11),T1,T2,ones(1,8),T2,T1,zeros(1,10)];
>>Hdr = [0,0,1,1]; wdl = [0,0.6,0.8,1];
>>k1 = 0:floor((M-1)/2); k2 = floor((M-1)/2)+1:M-1;
>>angH = [-alpha*(2*pi)/M*k1, alpha*(2*pi)/M*(M-k2)];
>>H = Hrs.*exp(j*angH);
>>h = real(ifft(H,M));
>>[db,mag,pha,grd,w] = freqz_m(h,1);
>>[Hr,ww,a,L] = Hr_Type1(h);
```

设计所得的时域和频域曲线在图 7.35 给出。

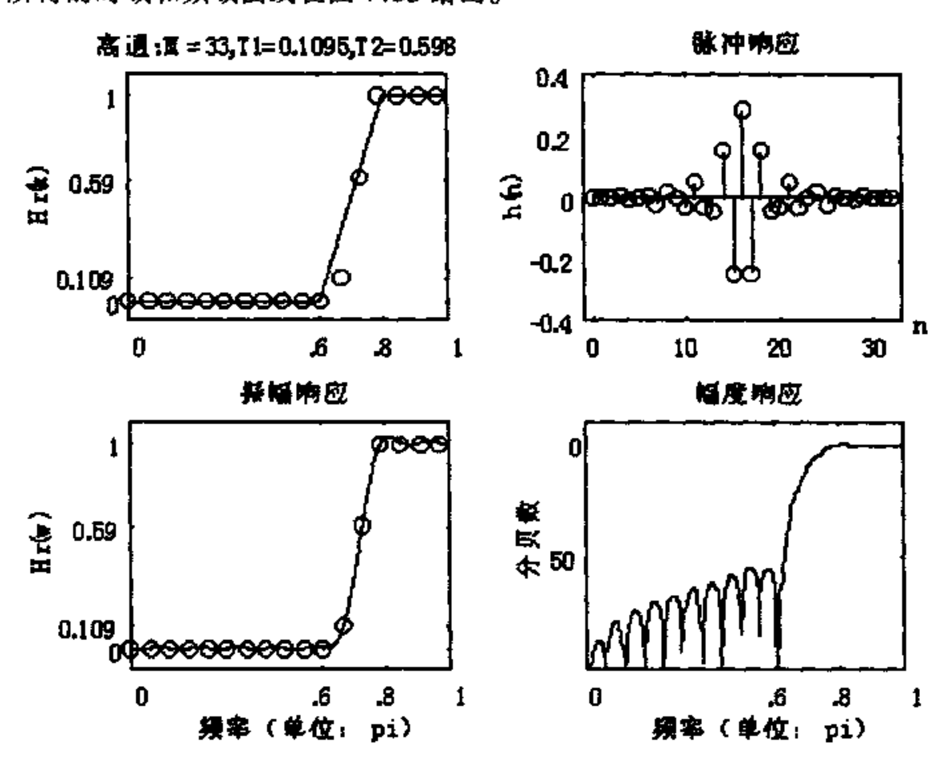


图 7.35　例 7.18 中的高通滤波器设计图

例 7.19　根据例 7.12 中式(7.31)给出的理想微分器，设计一个 33 点的数字微分器。

解：由(7.31)可知，(虚数)幅度响应的样本为：

$$jH_r(k)=\begin{cases}+j\dfrac{2\pi}{M}k, & k=0,\cdots,\left\lfloor\dfrac{M-1}{2}\right\rfloor\\ -j\dfrac{2\pi}{M}(M-k), & k=\left\lfloor\dfrac{M-1}{2}\right\rfloor+1,\cdots,M-1\end{cases}$$

对于线性相位，相位样本为：

$$\angle H(k)=\begin{cases}-\dfrac{M-1}{2}\dfrac{2\pi}{M}k=-\dfrac{M-1}{M}\pi k, & k=0,\cdots,\left\lfloor\dfrac{M-1}{2}\right\rfloor\\ +\dfrac{M-1}{M}\pi(M-k), & k=\left\lfloor\dfrac{M-1}{2}\right\rfloor+1,\cdots,M-1\end{cases}$$

因此

$$H(k)=jH_r(k)e^{j\angle H(k)},0\leqslant k\leqslant M-1 \qquad h(n)=\mathrm{IDFT}[H(k)]$$

```
>>M = 33; alpha = (M-1)/2; Dw = 2*pi/M;
>>l = 0:M-1; wl = Dw*l;
>>k1 = 0:floor((M-1)/2); k2 = floor((M-1)/2)+1:M-1;
>>Hrs = [j*Dw*k1, -j*Dw*(M-k2)];
>>angH = [-alpha*Dw*k1, alpha*Dw*(M-k2)];
>>H = Hrs.*exp(j*angH);
>>h = real(ifft(H,M));
>>[Hr,ww,a,P]=Hr_Type3(h);
```

时域和频域中的曲线如图 7.36 所示。要注意到，此微分器不是一个全通带微分器。

例 7.20 在理想希尔伯特变换器(7.32)基础上，设计一个 51 点的数字希尔伯特变换器。

解：由(7.32)得，(纯虚数)幅度响应的样本为：

$$jH_r(k)=\begin{cases}-j & k=1,\cdots,\left\lfloor\dfrac{M-1}{2}\right\rfloor\\ 0, & k=0\\ +j, & k=\left\lfloor\dfrac{M-1}{2}\right\rfloor+1,\cdots,M-1\end{cases}$$

由于这是一个 3 型线性相位滤波器，在 $\omega=\pi$ 处振幅响应为零。为了减小波动，在靠近 $\omega=\pi$处的过渡带中选择两个样本，其值在 0 与 j 之间。根据前面的经验，选择值为 $0.39j$。相位样本的选择与例 7.19 类似。

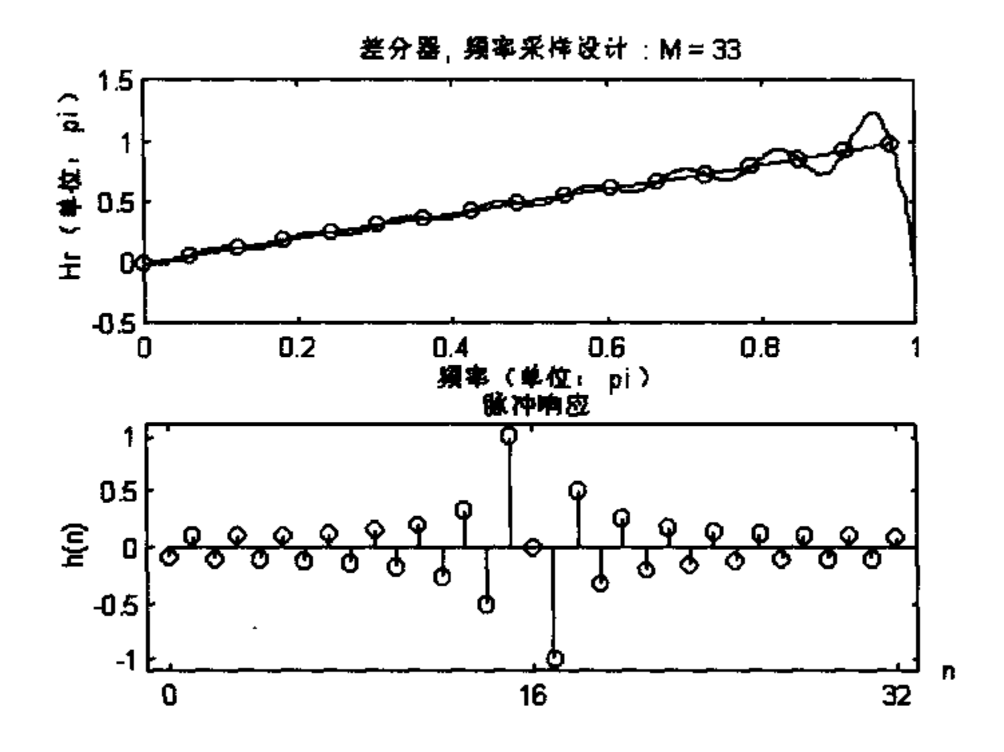


图 7.36 例 7.19 中的微分器设计图

```
>>M = 51; alpha = (M-1)/2; Dw = 2*pi/M;
>>l = 0:M-1; wl = Dw*l;
>>k1 = 0:floor((M-1)/2); k2 = floor((M-1)/2)+1:M-1;
>>Hrs = [0,-j*ones(1,(M-3)/2),-0.39j,0.39j,j*ones(1,(M-3)/2)];
>>angH = [-alpha*Dw*k1, alpha*Dw*(M-k2)];
>>H = Hrs.*exp(j*angH);
>>h = real(ifft(H,M));
>>[Hr,ww,a,P]=Hr_Type3(h);
```

图 7.37 给出过渡带样本对滤波器的影响。

以上讨论的频率采样滤波器叫做 A 型滤波器，其中采样频率为：

$$\omega_k = \frac{2\pi}{M}k, 0 \leqslant k \leqslant M-1$$

同时也存在第二种均匀分布的样本，他们由下式给出：

$$\omega_k = \frac{2\pi\left(k+\frac{1}{2}\right)}{M}, 0 \leqslant k \leqslant M-1$$

这叫做 B 型滤波器，在这里，频率采样结构同样是适用的。它的幅度响应 $H(e^{j\omega})$ 和脉冲响应

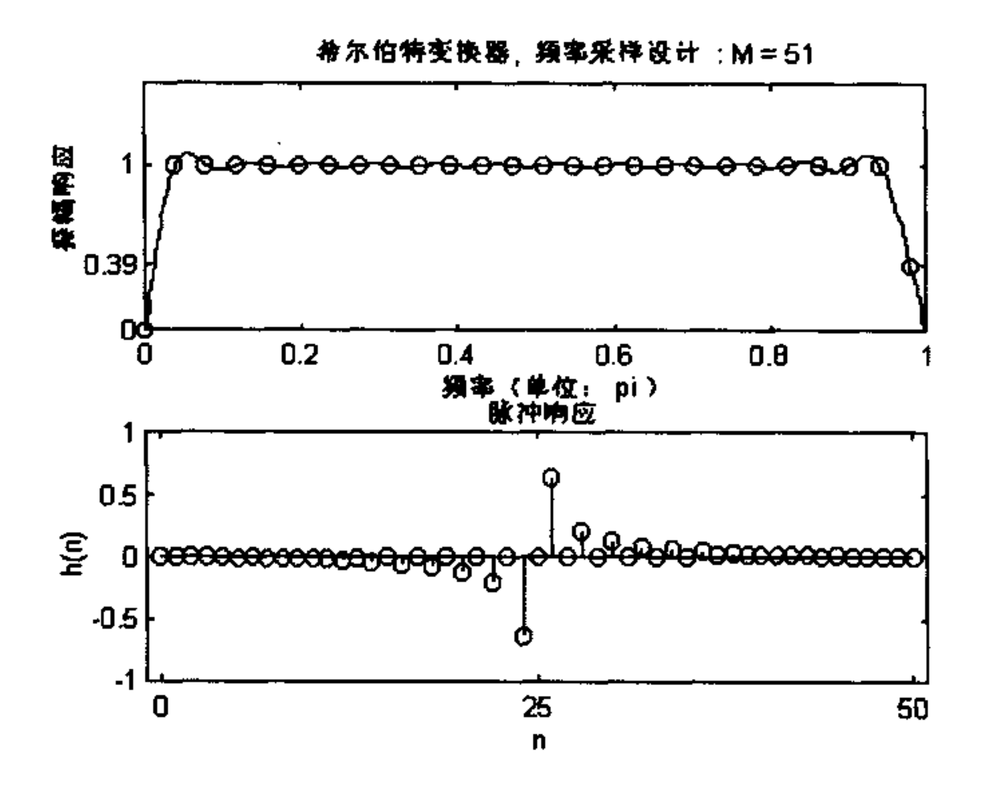


图 7.37　例 7.20 中的数字希尔伯特变换器设计图

要相对复杂一些，可从参考文献[19]中查得。也可以按本节讨论的方法用 MATLAB 实现这些设计。

最优等波动设计

上面探讨的两种技术，即窗设计和频率采样技术，都易于理解和实现。但是，他们存在一些不足：第一，在设计过程中，不能精确指定通带和阻带频率 ω_s 和 ω_p；无论设计所得的值是什么，都得接受。第二，不管是在窗设计方法中使 $\delta_1=\delta_2$，还是在频率采样法中只优化 δ_2，均不能同时确定波动系数 δ_1 和 δ_2。最后，逼近误差，也就是理想响应和实际响应的差，在带区间上不是均匀分布的。靠近带边缘处误差较高，而在远离边缘处较小，如果使误差均匀分布，就可以得到一个满足相同技术指标的较低阶滤波器。幸运的是，存在一种能克服上述三个问题的技术。相对来讲这种技术较难理解，并且它的实现需要计算机。

对于线性相位 FIR 滤波器，有可能导出一组条件，使得在最小化最大逼近误差（有时也叫做最大最小化或切比雪夫误差）的意义上说，设计是最优的。具有这种性质的滤波器叫做等波动滤波器，因为它在通带和阻带上的误差是均匀分布的，其阶次可以比较低。

下面首先阐述最大最小化最优 FIR 设计，并且讨论线性相位滤波器振幅响应的极大值和极小值（统称为极值）的总个数。然后讨论通用的等波动 FIR 滤波器设计算法，它用多项式内插来求解。这种算法叫做 Parks _ McClellan 算法，它要用到多项式求解的 Remez 交换程序。在许多计算机平台上，这种算法以子程序的形式存在。在本节中，我们将用 MATLAB 设计等波动 FIR 滤波器。

最大最小化问题的开发

在本章的前面，我们证明了四种情况下的线性相位 FIR 滤波器的频率响应可写为：

$$H(e^{j\omega}) = e^{j\beta}e^{-j\frac{M-1}{2}\omega}H_r(\omega)$$

其中，β 值和 $H_r(\omega)$ 的表达式在表 7.2 中给出。

表 7.2　线性相位 FIR 滤波器的振幅响应和 β 值

线性相位 FIR 滤波器类型	β	$H_r(e^{j\omega})$
1 型 M 为奇数 $h(n)$ 对称	0	$\sum_{0}^{(M-1)/2} a(n)\cos\omega n$
2 型 M 为偶数 $h(n)$ 对称	0	$\sum_{1}^{M/2} b(n)\cos[\omega(n-1/2)]$
3 型 M 为奇数 $h(n)$ 反对称	$\frac{\pi}{2}$	$\sum_{1}^{(M-1)/2} c(n)\sin\omega n$
4 型 M 为偶数 $h(n)$ 反对称	$\frac{\pi}{2}$	$\sum_{1}^{M/2} d(n)\sin[\omega(n-1/2)]$

表 7.3　线性相位 FIR 滤波器的 $Q(\omega)$，L 和 $P(\omega)$

LP FIR 滤波器类型	$Q(\omega)$	L	$P(\omega)$
1 型	1	$\frac{M-1}{2}$	$\sum_{0}^{L} a(n)\cos\omega n$
2 型	$\cos\frac{\omega}{2}$	$\frac{M}{2}-1$	$\sum_{0}^{L} \tilde{b}(n)\cos\omega n$
3 型	$\sin\omega$	$\frac{M-3}{2}$	$\sum_{0}^{L} \tilde{c}(n)\cos\omega n$
4 型	$\sin\frac{\omega}{2}$	$\frac{M}{2}-1$	$\sum_{0}^{L} \tilde{d}(n)\cos\omega n$

利用简单的三角关系，可把上面的每个 $H_r(\omega)$ 改写成 ω 的固定函数（称它为 $Q(\omega)$）和一个余弦的和函数（叫做 $P(\omega)$）的乘积，详细讨论见[19]和习题 7.1－7.4。即：

$$H_r(\omega) = Q(\omega)P(\omega) \tag{7.40}$$

其中 $P(\omega)$ 的形式为：

$$P(\omega) = \sum_{n=0}^{L} \alpha(n)\cos\omega n \tag{7.41}$$

四种情况下的 $Q(\omega)$，L 和 $P(\omega)$ 在表 7.3 中给出。

上述分析的目的是为了得到四种情况下 $H_r(\omega)$ 的通用形式，它使得问题的阐述更容易一些。为了把我们的问题表示成一个切比雪夫逼近问题，首先必须定义期望的振幅响应 $H_{dr}(\omega)$ 和一个权函数 $W(\omega)$，两者均定义在通带及阻带上。权函数对于独立控制 δ_1 和 δ_2 是十分必

要的。加权误差定义为：

$$E(\omega) \triangleq W(\omega)[H_{dr}(\omega) - H_r(\omega)], \omega \in S \triangleq [0,\omega_p] \cup [\omega_s,\pi] \tag{7.42}$$

下面的曲线清楚地说明了这些概念，它给出了典型的等波动滤波器响应和它的理想响应。

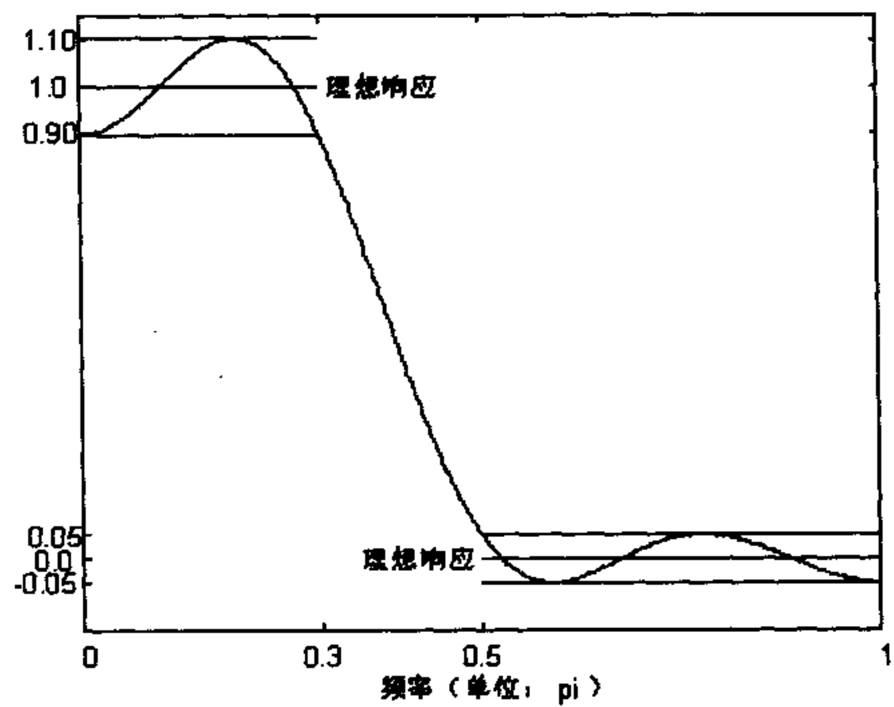


图 7.38 等波动滤波器的振幅响应

下图给出误差响应$[H_{dr}(\omega) - H_r(\omega)]$。

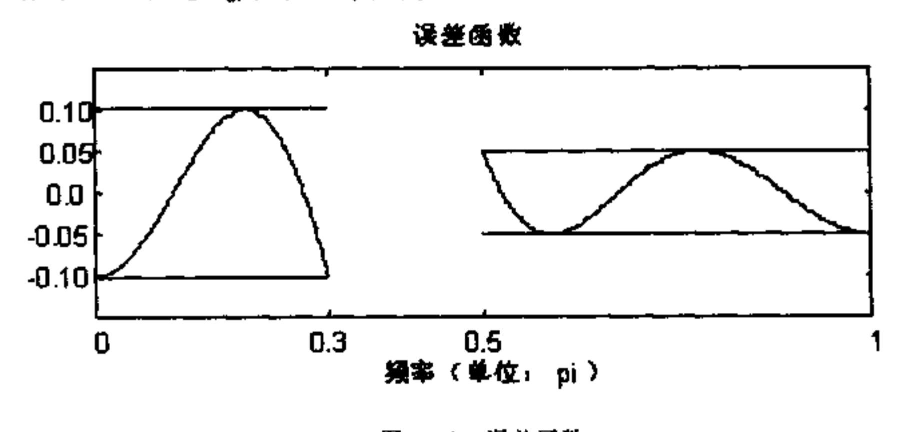


图 7.39 误差函数

如果选择

$$W(\omega)=\begin{cases}\dfrac{\delta_2}{\delta_1}, & 在通带中\\ 1, & 在阻带中\end{cases} \tag{7.43}$$

那么加权误差响应 $E(\omega)$为：

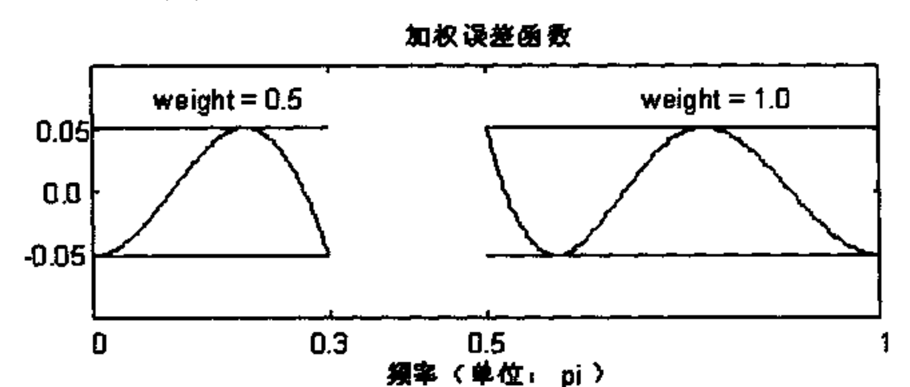


图 7.40　加权误差函数

由此看出，通带和阻带上的最大误差均为 δ_2。如果我们成功地把最大加权误差最小化至 δ_2，则也自动满足了通带上的技术指标 δ_1。把(7.40)中的 $H_r(\omega)$代入(7.42)，得：

$$\begin{aligned}E(\omega)&=W(\omega)[H_{dr}(\omega)-Q(\omega)P(\omega)]\\&=W(\omega)Q(\omega)\left[\frac{H_{dr}(\omega)}{Q(\omega)}-P(\omega)\right],\omega\in S\end{aligned}$$

如果定义

$$\hat{W}(\omega)\triangleq W(\omega)Q(\omega) \qquad \text{and} \qquad \hat{H}_{dr}(\omega)\triangleq\frac{H_{dr}(\omega)}{Q(\omega)}$$

得到

$$E(\omega)=\hat{W}(\omega)[\hat{H}_{dr}(\omega)-P(\omega)],\omega\in S \tag{7.44}$$

由此，我们得到了四种情况下 $E(\omega)$的通用形式。

问题叙述　切比雪夫逼近问题可定义为：

确定系数 $a(n)$或 $\tilde{b}(n)$ 或 $\tilde{c}(n)$ 或 $\tilde{d}(n)$ [或者等价地求出 $a(n)$或 $b(n)$或 $c(n)$或 $d(n)$]，使通带和阻带上的最大绝对误差 $E(\omega)$最小，即，

$$\min_{\text{over. coeff.}}\left[\max_{\omega\in S}|E(\omega)|\right] \tag{7.45}$$

现在已成功地确定了精确的 ω_p、ω_s、δ_1 和 δ_2。并且其误差在通带和阻带上是均匀分布的。

对极值数目的限制

在对上述问题求解之前，首先讨论对于给定的 M 点滤波器，误差函数 $E(\omega)$有多少个局部极大值和极小值？在 Parks-McClellan 算法中，为了得到多项式内插要用到此信息。答案在 $P(\omega)$表达式中。根据(7.41)，$P(\omega)$是 ω 的三角函数，利用三角关系式

$$
\begin{aligned}
\cos(2\omega) &= 2\cos(\omega) - 1 \\
\cos(3\omega) &= 4\cos^3(\omega) - 3\cos(\omega) \\
\vdots &= \vdots
\end{aligned}
$$

$P(\omega)$可转换成一个 $\cos(\omega)$的三角多项式，把(7.41)改写为

$$
P(\omega) = \sum_{n=0}^{L} \beta(n)\cos^n\omega \tag{7.46}
$$

例 7.21 设 $h(n) = [1,2,3,4,3,2,1]/15$，则 $M = 7$，而且 $h(n)$是对称的，因此是 1 型线性相位滤波器。且 $L = (M-1)/2 = 3$。根据(7.7)

$$
\alpha(n) = a(n) = 2h(3-n), 1 \leqslant n \leqslant 2; \qquad \text{and} \qquad \alpha(0) = a(0) = h(3)
$$

或 $\alpha(n) = \frac{1}{15}[4,6,4,2]$，因此

$$
\begin{aligned}
P(\omega) &= \sum_{0}^{3} \alpha(n)\cos\omega n = \frac{1}{15}(4 + 6\cos\omega + 4\cos2\omega + 2\cos3\omega) \\
&= \frac{1}{15}\{4 + 6\cos\omega + 4(2\cos^2\omega - 1) + 2(4\cos^3\omega - 3\cos\omega)\} \\
&= 0 + 0 + \frac{8}{15}\cos^2\omega + \frac{8}{15}\cos^3\omega = \sum_{0}^{3}\beta(n)\cos^n\omega
\end{aligned}
$$

或者 $\beta(n) = [0,0,\frac{8}{15},\frac{8}{15}]$

从(7.46)可知，$P(\omega)$是 $\cos(\omega)$的 L 阶多项式。由于 $\cos(\omega)$在开区间 $0 < \omega < \pi$ 是一个单调函数，因此它应该与一般的 x 的 L 阶多项式$P(x)$类似，因此 $P(\omega)$在开区间 $0 < \omega < \pi$ 上最多(即不多于)有$(L-1)$个极值。例如

$$
\cos^2(\omega) = \frac{1 + \cos2\omega}{2}
$$

仅在 $\omega = \pi/2$ 处，有一最小值。然而，在闭区间 $0 \leqslant \omega \leqslant \pi$ 上它有三个极值(即，$\omega = 0$ 时为最大

值，$\omega=\pi/2$ 时为最小值，$\omega=\pi$ 时为最大值）。如果包括端点 $\omega=0$ 和 $\omega=\pi$，则在闭区间 $0\leqslant\omega\leqslant\pi$ 上 $P(\omega)$ 最多有 $(L+1)$ 个局部极值。我们还希望滤波器的技术指标在带边缘 ω_p 和 ω_s 上完全符合要求。因此在 $0\leqslant\omega\leqslant\pi$ 上，能满足技术指标要求的极值频率不超过 $(L+3)$ 个。

结论 在该区域中，误差函数 $E(\omega)$ 最多有 $(L+3)$ 个极值。

例 7.22 画出例7.21中滤波器的振幅响应，并且求出相应的误差函数中的极值个数。

解：脉冲响应为：

$$h(n)=[1,2,3,4,3,2,1]/15,\ M=7\ \text{或}\ L=3$$

并且，从例 7.21 知道 $\alpha(n)=\frac{1}{15}[4,6,4,2]$ 和 $\beta(n)=[0,0,\frac{8}{15},\frac{8}{15}]$，因此

$$P(\omega)=\frac{8}{15}\cos^2\omega+\frac{8}{15}\cos^3\omega$$

如图 7.41 所示。显然，$P(\omega)$ 在开区间 $0<\omega<\pi$ 上有 $(L-1)=2$ 个极值。图中也给出了误差函数，它有 $(L+3)=6$ 个极值点。

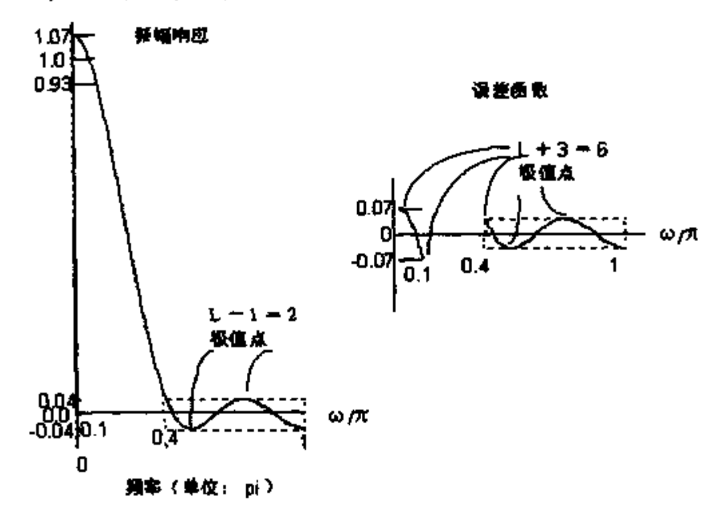


图 7.41 例 7.22 中的振幅响应和误差函数

现在，我们把注意力转向问题(7.45)，在逼近理论中，这是一个为人熟知的问题，可用下面的重要定理求出它的解。

定理 1 交替定理

设 S 是闭区间 $[0,\pi]$ 上的任意闭子集，为了使 $P(\omega)$ 是 $H_{dr}(\omega)$ 在 S 上唯一的最大最小化逼近，必须且只须误差函数 $E(\omega)$ 在 S 上至少有 $(L+2)$ 个"交替点"或极值频率，也就是说，S 中必须有 $(L+2)$ 个频率 ω_i 满足

$$E(\omega_i)=-E(\omega_{i-1})=\pm\max_S|E(\omega)|$$

$$\triangleq\pm\delta,\ \forall\,\omega_0<\omega_1<\cdots<\omega_{L+1}\in S$$

$$(7.47)$$

把此定理与前面的讨论相结合可以推出,最优等波动滤波器的误差函数在 S 上有($L+2$)或($L+3$)个极值。大多数等波动滤波器有($L+2$)个极值。但是对于某些 ω_p、ω_s 的组合,可能得到有($L+3$)个极值的滤波器。它们的响应中有一个附加的波动,因此叫做附加波动滤波器。

Parks-McClellan 算法

交替定理保证最大最小逼近问题的解存在并且唯一,但它并没有说明怎样得到这个解。我们既不知道长度 M(或 L),也不知道极值频率 ω_i、参数 $\{a(n)\}$ 和最大误差 δ。Parks 和 McClellan [17]提供了利用 Remez 交换算法导出的迭代算法。它假设已知滤波器长度 M(或 L)和比率 δ_1/δ_2。如果按(7.43)选择了权函数,也正确地选择了阶数 M,并且设 $\delta=\delta_2$。这时,就可得到解。显然,δ 和 M 是相关的,M 越大,δ 越小。滤波器技术指标中给出了 δ_1、δ_2、ω_p 和 ω_s,因此需要设定 M 的值。凯泽提出了一个简单的公式来逼近 M。它是:

$$\hat{M} = \frac{-20\log_{10}\sqrt{\delta_1\delta_2}-13}{14.6\Delta f}+1;\Delta f = \frac{\omega_s-\omega_p}{2\pi} \tag{7.48}$$

Parks-McClellan 算法首先猜设($L+2$)个极值频率 $\{\omega_i\}$,估计这些频率上的最大误差,接着按(7.47)给定的各点,拟合一个 L 阶多项式(7.46)。然后在一个较细的网格上确定局部极大误差及其极值频率 $\{\omega_i\}$,由这些新频率点拟合出一个新的 L 阶多项式,重复以上过程。一直进行至找到最优集 $\{\omega_i\}$ 和全局最大误差 δ 为止。此迭代过程保证是收敛的,从而得到多项式 $P(\omega)$。根据(7.46)求出系数 $\beta(n)$。最后算出系数 $a(n)$ 和脉冲响应 $h(n)$。在 MATLAB 中,此算法的形式为 remez 函数,这会在下面说明。

由于 M 是近似的,最大误差 δ 可能不等于 δ_2。如果出现这种情况,需要增加 M(若 $\delta>\delta_2$)或减小 M(若 $\delta<\delta_2$),再次用 remez 算法确定一个新的 δ。重复此过程至 $\delta\leqslant\delta_2$。这样就得到了等波动滤波器,它满足前面讨论的三个要求。

MATLAB 实现

在 MATLAB 中,实现 Parks-McClellan 算法的函数为 remez,它最常用的句法为:

[h] = remez(N,f,m,weights,ftype)

它有几种调用形式:*

·[h] = remez(N,f,m)设计一个 N 阶(滤波器的长度为 $M=N+1$)FIR 数字滤波器,它的频率响应在数组 f 和 m 中给定。长度为 M 的数组 h 返回的滤波器系数(或脉冲响应)。数组 f 中包含边缘频率,其单位为 π,即 $0.0\leqslant f\leqslant 1.0$。这些频率必须以递增次序排列,从 0.0 开始,到 1.0 结束。数组 m 为各指定频率上预期的幅度响应。f 和 m 的长度必须相等且为偶数,每个带中所用的权函数等于 1,这说明在每个带中的容限(δ_i)是相同的。

·[h] = remez(N,f,m,weights) 与上述情形相似,数组 weights 是每个带的权函数。

* 从老的 MATLAB 学生版本到新的学生版本,(或从数字信号处理工具箱 2.0b 到版本 3.0)remez 函数有一些小的变化,这里的叙述符合新版本。

·[h] = remez(N,f,m,ftype)与第一种情形类似,除了当 ftype 是字符串"hilbert"或"differentiator"时,它相应地设计数字希尔伯特变换器或数字微分器。对于数字希尔伯特变换器来说,数组 f 中的最低频率不能等于 0,最高频率不能为 1。对于数字微分器,矢量 m 不给出每个带中预期的斜率,而是给出预期的幅度。

·[h] = remez(N,f,m,weights,ftype)与上一种情况类似,只是由数组 weights 指定每个带的权函数。

正如在讨论 Parks-McClellan 算法时指出的,为了使用程序 remez,首先必须根据(7.48)假设滤波器的阶数,得到数组 h 中的滤波器系数后,还必须检查最小阻带衰减,并与给定的 As 比较,然后增加(或减小)滤波器的阶数。重复此过程直到得到期望的 As。我们用下面几个 MATLAB 例子来说明此过程。

例 7.23 利用 Parks-McClellan 算法设计例 7.8 中的低通滤波器。设计参数为:

$$\omega_p = 0.2\pi,\ R_p = 0.25\text{dB}$$
$$\omega_s = 0.3\pi,\ A_s = 50\ \text{dB}$$

解:设计此滤波器的 MATLAB 程序为:

```
>>wp = 0.2*pi; ws = 0.3*pi; Rp = 0.25; As = 50;
>>wsi = ws/delta-w+1;
>>delta1 = (10^(Rp/20)-1)/(10^(Rp/20)+1)
>>delta2 = (1+delta1)*(10^(-As/20))
>>deltaH = max(delta1,delta2); deltaL = min(delta1,delta2);
>>weights = [delta2/delta1 1]
>>deltaf = (ws-wp)/(2*pi)
>>M = ceil((-20*log10(sqrt(delta1*delta2))-13)/(14.6*deltaf)+1)
M = 43
>>f = [0 wp/pi ws/pi 1]
>>m = [1 1 0 0];
>>h = remez(M-1,f,m,weights);
>>[db,mag,pha,grd,w] = freqz_m(h,[1]);
>>delta_w = 2*pi/1000;
>>Asd = -max(db(wsi:1:501))
Asd = 47.8562
>>M = M+1;
>>h = remez(M-1,f,m,weights);
>>[db,mag,pha,grd,w] = freqz_m(h,[1]);
>>Asd = -max(db(wsi:1:501))
Asd = 48.2155
>>M = M+1;
>>h = remez(M-1,f,m,weights);
```

```
>>[db,mag,pha,grd,w] = freqz_m(h,[1]);
>>Asd = -max(db(wsi:1:501))
Asd = 48.8632
>>M = M+1;
>>h = remez(M-1,f,m,weights);
>>[db,mag,pha,grd,w] = freqz_m(h,[1]);
>>Asd = -max(db(wsi:1:501))
Asd = 49.8342
M = M+1;
>>h = remez(M-1,f,m,weights);
>>[db,mag,pha,grd,w] = freqz_m(h,[1]);
>>Asd = -max(db(wsi:1:501))
Asd = 51.0896
>>M
M = 47
```

当算得的阻带衰减超过给定的 A_s 时，终止迭代过程，找到的最优的 M 值为 47。这个值比采用窗设计技术(凯泽窗的 $M=61$)或频率采样技术($M=60$)要小得多。图 7.42 给出此滤波器在时域和频域中的响应曲线及其在通带及阻带中的误差函数，以说明等波动特性。

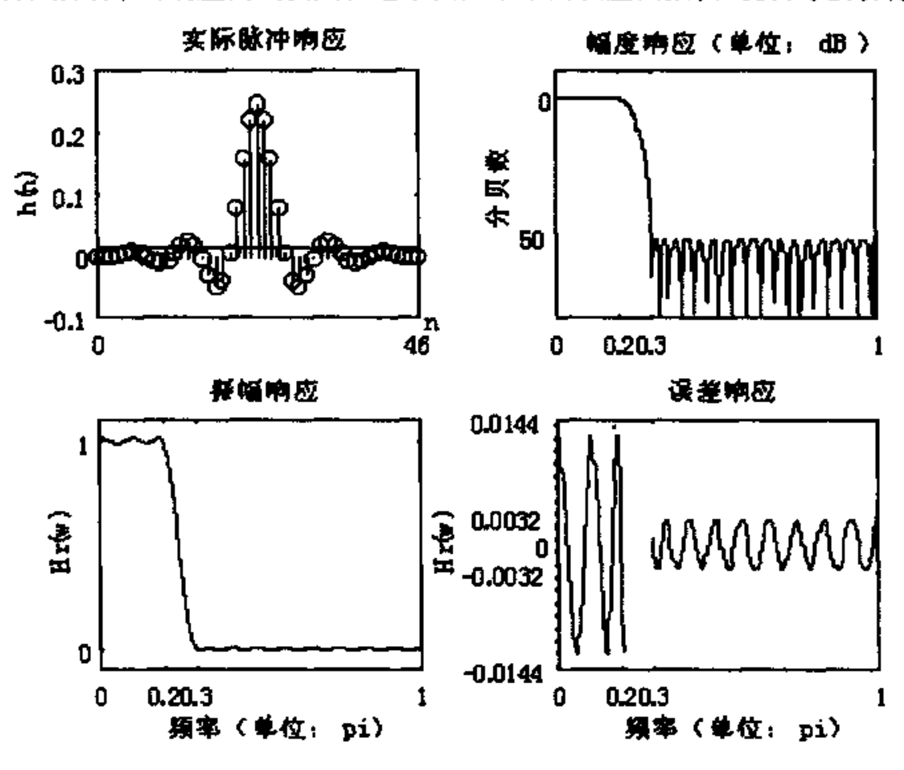


图 7.42　例 7.23 中等波动低通 FIR 滤波器的响应图

例 7.24　利用 Parks-McClellan 算法设计例 7.10 中的带通滤波器。设计参数为：

$$
\begin{aligned}
\omega_{1s} &= 0.2\pi, \\
\omega_{1p} &= 0.35\pi, R_p = 1\ \text{dB} \\
\omega_{2p} &= 0.65\pi, \\
\omega_{2s} &= 0.8\pi,
\end{aligned}
\quad A_s = 60\ \text{dB}
$$

解:下面的 MATLAB 程序说明设计此滤波器的方法:

```
>>ws1 = 0.2*pi; wp1 = 0.35*pi; wp2 = 0.65*pi; ws2 = 0.8*pi;
>>wsli=floor(ws1/delta_w)+1;
>>Rp = 1.0; As = 60;
>>delta1 = (10^(Rp/20)-1)/(10^(Rp/20)+1);
>>delta2 = (1+delta1)*(10^(-As/20));
>>deltaH = max(delta1,delta2); deltaL = min(delta1,delta2);
>>weights = [1 delta2/delta1 1];
>>delta_f =min((ws2-wp2)/(2*pi), (wp1-ws1)/(2*pi));
>>M = ceil((-20*log10(sqrt(delta1*delta2))-13)/(14.6*delta_f)+1)
M = 28
>>f = [0 ws1/pi wp1/pi wp2/pi ws2/pi 1];
>>m = [0 0 1 1 0 0];
>>h = remez(M-1,f,m,weights);
>>[db,mag,pha,grd,w] = freqz_m(h,[1]);
>>delta_w=2*pi/1000;
>>Asd = -max(db(1:1:wsli))
Asd = 56.5923
>>M = M+1;
>>h = remez(M-1,f,m,weights);
>>[db,mag,pha,grd,w] = freqz_m(h,[1]);
>>Asd = -max(db(1:1:ws1/delta_w))
Asd = 61.2818
>>M = M+1;
>>h = remez(M-1,f,m,weights);
>>[db,mag,pha,grd,w] = freqz_m(h,[1]);
>>Asd = -max(db(1:1:ws1/delta_w))
Asd = 60.3820
>>M = M+1;
>>h = remez(M-1,f,m,weights);
>>[db,mag,pha,grd,w] = freqz_m(h,[1]);
>>Asd = -max(db(1:1:ws1/delta_w))
Asd = 61.3111
```

```
>>M
M = 31
```

M 的最优值为 31，设计的滤波器在时域和频域中的响应图如图 7.43 所示。

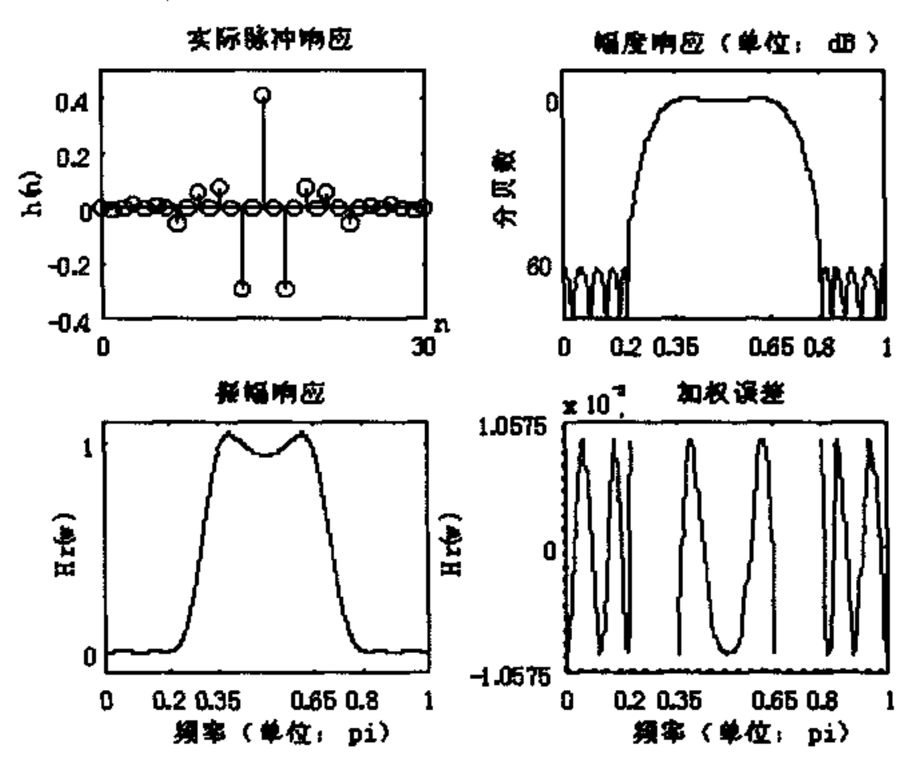


图 7.43　例 7.24 中等波动带通 FIR 滤波器的响应图

例 7.25　设计一个具有下列技术指标的高通滤波器：

$$\omega_s = 0.6\pi, \qquad A_s = 50 \text{ dB}$$
$$\omega_p = 0.75\pi, \qquad R_p = 0.5\text{dB}$$

解：由于这是一个高通滤波器，必须保证 M 为奇数。下面给出 MATLAB 程序。

```
>>ws = 0.6*pi; wp = 0.75*pi; Rp = 0.5; As = 50;
>>delta1 = (10^(Rp/20)-1)/(10^(Rp/20)+1);
>>delta2 = (1+delta1)*(10^(-As/20));
>>deltaH = max(delta1,delta2); deltaL = min(delta1,delta2);
>>weights = [1 delta2/delta1];
>>deltaf = (wp-ws)/(2*pi);
>>M = ceil((-20*log10(sqrt(delta1*delta2))-13)/(14.6*deltaf)+1);
% M 必须为奇数
>>M = 2*floor(M/2)+1
M = 27
```

```
>>f = [0 ws/pi wp/pi 1]
>>m = [0 0 1 1];
>>h = remez(M-1,f,m,weights);
>>[db,mag,pha,grd,w] = freqz_m(h,[1]);
>>delta_w = 2*pi/1000; wsi = ws/delta_w; wpi = wp/delta_w;
>>Asd = -max(db(1:1:wsi))
Asd = 49.5918
>>M = M+2;        % M 必须为奇数
>>h = remez(M-1,f,m,weights);
>>[db,mag,pha,grd,w] = freqz_m(h,[1]);
>>Asd = -max(db(1:1:wsi))
Asd = 50.2253
>>M
M = 29
```

注意增加 M 时,保持它为奇数。M 的最优值为 37,设计的滤波器在时域和频域中的响应图如图 7.44 所示。

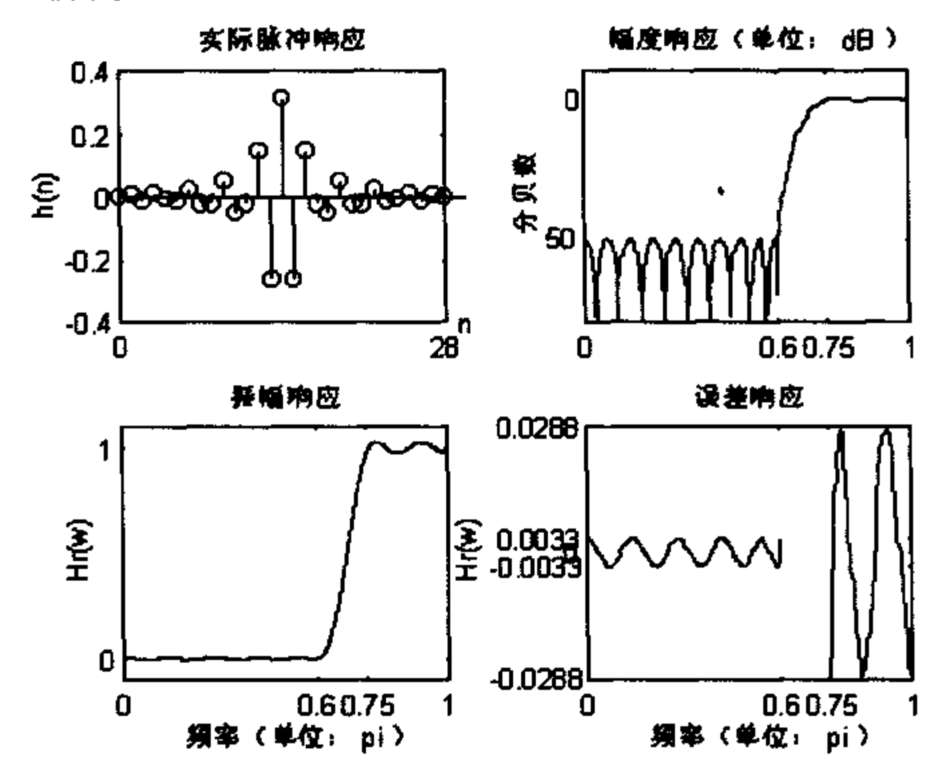


图 7.44 例 7.25 的等波动高通 FIR 滤波器的响应图

例 7.26 这个例子设计一个“阶梯”滤波器,它分为三段,每段有不同的理想响应和容限,设计技术指标为:

第一段: $0\leqslant\omega\leqslant0.3\pi$ 理想增益 = 1, 容限 $\delta_1=0.01$

第二段: $0.4\pi\leqslant\omega\leqslant0.7\pi$ 理想增益 = 0.5, 容限 $\delta_2=0.005$

第三段：$0.8\pi \leqslant \omega \leqslant \pi$　　　　理想增益 = 0，　　　　容限 $\delta_3 = 0.001$

解：下面的 MATLAB 程序给出设计过程。

```
>>w1 = 0; w2 = 0.3*pi; delta1 = 0.01;
>>w3 = 0.4*pi; w4 = 0.7*pi; delta2 = 0.005;
>>w5 = 0.8*pi; w6 = pi; delta3 = 0.001;
>>weights = [delta3/delta1 delta3/delta2 1]
>>delta_f = min((w3-w2)/(2*pi), (w5-w3)/(2*pi))
>>M = ceil((-20*log10(sqrt(delta1*delta2))-13)/(14.6*delta_f)+1)
M = 43
>>f = [0 w2/pi w3/pi w4/pi w5/pi 1];
>>m = [1 1 0.5 0.5 0 0];
>>h = remez(M-1,f,m,weights);
>>[db,mag,pha,grd,w] = freqz_m(h,[1]);
>>delta_w = 2*pi/680;
>>Asd = -max(db(w5i/delta_w+10:1:341))
Asd = 56.2181
```

M = 49 时找到最优值。

```
>>M = 49;
>>h = remez(M-1,f,m,weights);
>>[db,mag,pha,grd,w] = freqz_m(h,[1]);
>>Asd = -max(db(w5/delta_w+10:1:341))
Asd = 60.6073
```

设计的滤波器在时域和频域中的响应图如图 7.45 所示。

例 7.27　本例设计一个数字微分器，它在每段上具有不同的斜率。技术指标为：

第一段：$0 \leqslant \omega \leqslant 0.2\pi$　　　　斜率 = 1 个 样本/周期

第二段：$0.4\pi \leqslant \omega \leqslant 0.6\pi$　　　　斜率 = 2 个样本/周期

第三段：$0.8\pi \leqslant \omega \leqslant \pi$　　　　斜率 = 3 个样本/周期

解：我们需要每段上的幅度响应值，这可由带边缘频率（单位为周数/样本）与斜率（单位为样本/周）相乘得到

第一段：$0 \leqslant f \leqslant 0.1$　　　斜率 = 1 个 样本/周 $\Rightarrow 0.0 \leqslant |H| \leqslant 0.1$

第二段：$0.2 \leqslant f \leqslant 0.3$　　　斜率 = 2 个样本/周 $\Rightarrow 0.4 \leqslant |H| \leqslant 0.6$

第三段：$0.4 \leqslant f \leqslant 0.5$　　　斜率 = 3 个样本/周 $\Rightarrow 1.4 \leqslant |H| \leqslant 1.5$

设所有段上的权相等，则 MATLAB 程序为：

```
>>f = [0 0.2 0.4 0.6 0.8 1];          % 单位：w/pi
>>m = [0,0.1,0.4,0.6,1.2,1.5];        % 幅度矢量
```

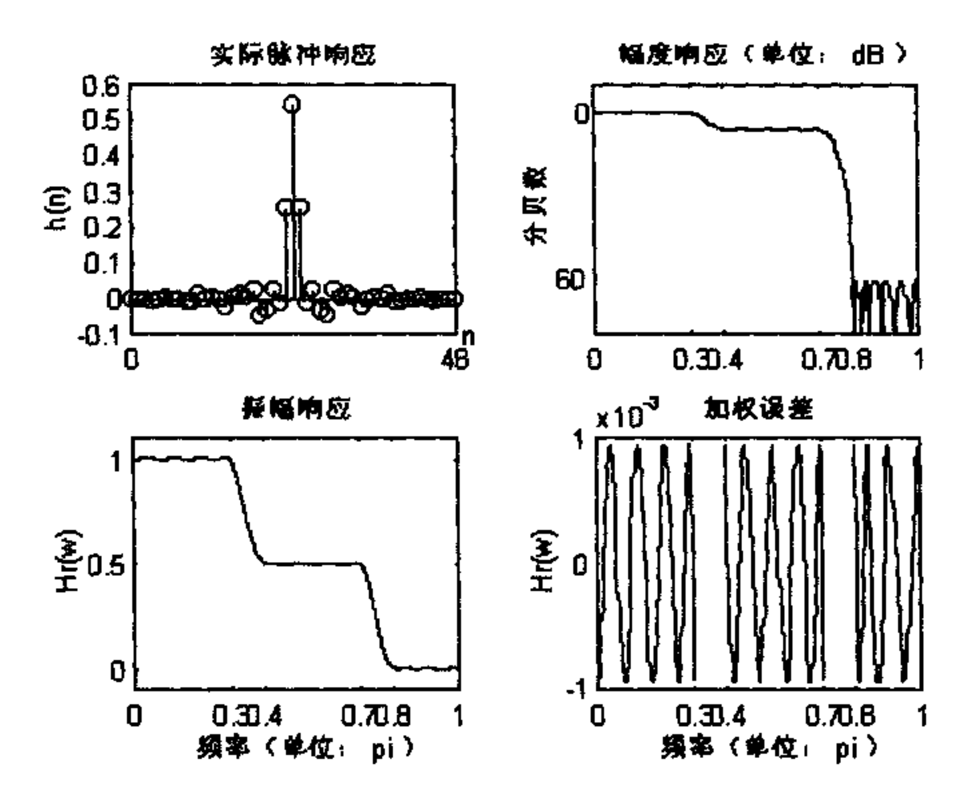


图 7.45　例 7.26 的等波动阶梯 FIR 滤波器的响应图

```
>>h = remez(25,f,m,'differentiator');
>>[db,mag,pha,grd,w] = freqz_m(h,[1]);
>>figure(1); subplot(1,1,1)
>>subplot(2,1,1); stem([0:25],h); title('脉冲响应');
>>xlabel('n');ylabel('h(n)'); axis([0,25,-0.6,0.6]);
>>set(gca,'XTickMode','manual','XTick',[0,25])
>>set(gca,'YTickMode','manual','YTick',[-0.6:0.2:0.6]);
>>subplot(2,1,2); plot(w/(2*pi),mag); title('幅度响应')
>>xlabel('归一化频率 f'); ylabel('|H|')
>>set(gca,'XTickMode','manual','XTick',f/2)
>>set(gca,'YTickMode','manual','YTick',m);grid
```

图 7.46 给出频域中的响应图。

例 7.28　在频段$0.05\pi \leqslant \omega \leqslant 0.95\pi$上设计一个希尔伯特变换器。

解:这是一个宽带希尔伯特变换器,应当选择奇数长度(即 3 型滤波器),取 $M=51$。

```
>>f = [0.05,0.95]; m = [1 1]; M = 51; N = M-1;
>>h = remez(N,f,m,'hilbert');
>>[db,mag,pha,grd,w] = freqz_m(h,[1]);
>>figure(1); subplot(1,1,1)
>>subplot(2,1,1); stem([0:N],h); title('脉冲响应');
```

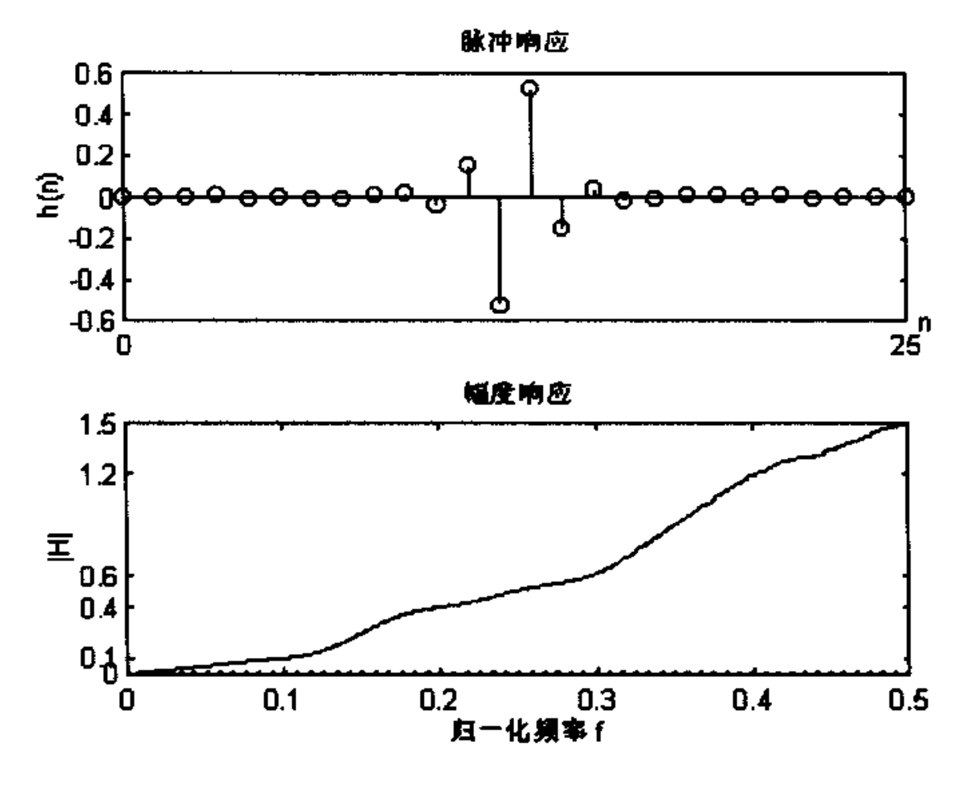


图 7.46 例 7.27 的微分器响应图

```
> > xlabel('n');ylabel('h(n)'); axis([0,N, -0.8,0.8]);
> > set(gca,'XTickMode','manual','XTick',[0,N])
> > set(gca,'YTickMode','manual','YTick',[-0.8:0.2:0.8]);
> > subplot(2,1,2); plot(w/pi,mag); title('幅度响应')
> > xlabel('频率(单位:pi)'); ylabel('|H|')
> > set(gca,'XTickMode','manual','XTick',[0,f,1])
> > set(gca,'YTickMode','manual','YTick',[0,1]);grid
```

图 7.47 给出希尔伯特变换器的响应图。

习 题

P7.1 1 型线性相位滤波器由下式表征：

$$h(n) = h(M-1-n), \qquad 0 \leqslant n \leqslant M-1, M\text{ 为奇数}$$

证明它的振幅响应 $H_r(\omega)$为：

$$H_r(\omega) = \sum_{n=0}^{L} a(n)\cos\omega n, L = \frac{M-1}{2}$$

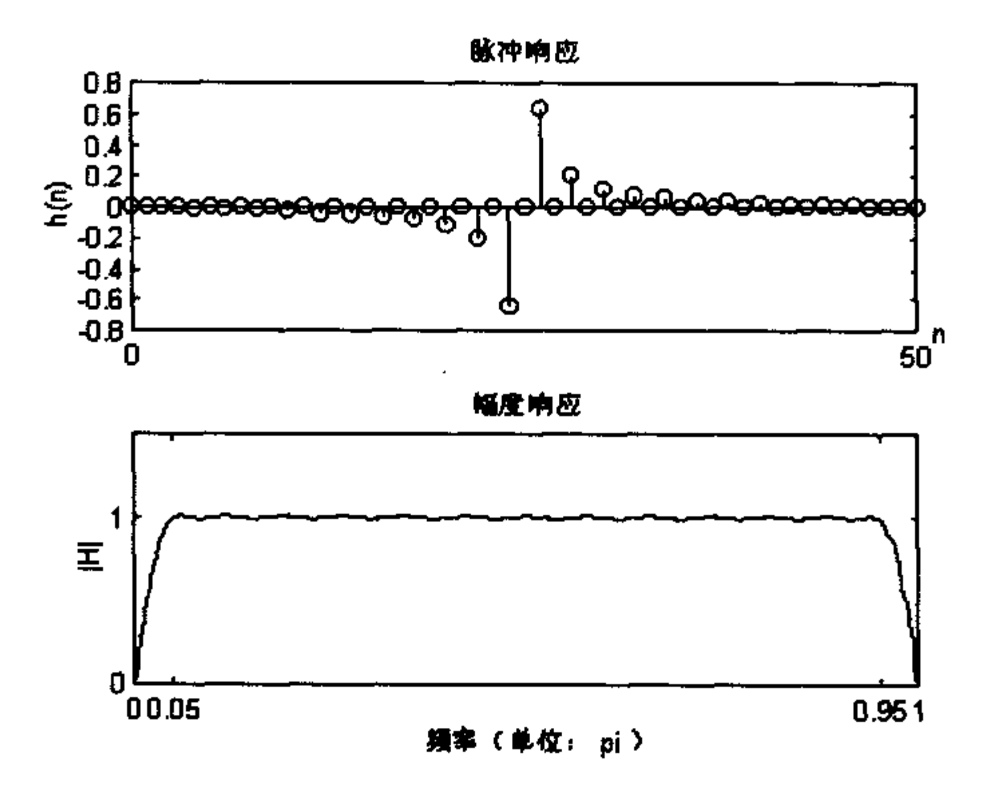


图 7.47 例 7.28 的希尔伯特变换器响应图

其中系数$\{a(n)\}$是从 $h(n)$得到的。

P7.2 2 型线性相位滤波器由下式表征：

$$h(n)=h(M-1-n),\qquad 0\leqslant n\leqslant M-1, M \text{ 为偶数}$$

a.证明它的振幅响应 $H_r(\omega)$为：

$$H_r(\omega)=\sum_{n=1}^{M/2}b(n)\cos\left\{\omega\left(n-\frac{1}{2}\right)\right\}$$

其中系数$\{b(n)\}$是从 $h(n)$得到的。

b. 证明上面的 $H_r(\omega)$可进一步表示成：

$$H_r(\omega)=\cos\frac{\omega}{2}\sum_{n=0}^{L}\tilde{b}(n)\cos\omega n, L=\frac{M}{2}-1$$

其中 $\tilde{b}(n)$由 $b(n)$导出。

P7.3 3 型线性相位滤波器由下式表征：

$$h(n)=-h(M-1-n),\qquad 0\leqslant n\leqslant M-1, M \text{ 为奇数}$$

a.证明它的振幅响应 $H_r(\omega)$为:

$$H_r(\omega) = \sum_{n=1}^{M/2} c(n)\sin\omega n$$

其中系数$\{c(n)\}$是从 $h(n)$得到的。

b. 证明上面的 $H_r(\omega)$可进一步表示成:

$$H_r(\omega) = \sin\omega \sum_{n=0}^{L} \tilde{c}(n)\cos\omega n, L = \frac{M-3}{2}$$

其中 $\tilde{c}(n)$由 $c(n)$导出。

P7.4 4型线性相位滤波器由下式表征:

$$h(n) = -h(M-1-n), \qquad 0 \leqslant n \leqslant M-1, M \text{ 为偶数}$$

a.证明它的振幅响应 $H_r(\omega)$为:

$$H_r(\omega) = \sum_{n=1}^{M/2} d(n)\sin\left\{\omega\left(n - \frac{1}{2}\right)\right\}$$

其中系数$\{d(n)\}$是从 $h(n)$得到的。

b. 证明上面的 $H_r(\omega)$可进一步表示成:

$$H_r(\omega) = \sin\frac{\omega}{2} \sum_{n=0}^{L} \tilde{d}(n)\cos\omega n, L = \frac{M}{2} - 1$$

其中 $\tilde{d}(n)$由 $d(n)$导出。

P7.5 写出一个 MATLAB 函数,当给定线性相位脉冲响应 $h(n)$时,用它来计算振幅响应 $H_r(\omega)$。

其格式应为:

```
function [Hr,w,P,L] = Ampl_Res(h);
%
%function [Hr,w,P,L] = Ampl_Res(h)
% 计算振幅响应 Hr(ω)和它阶数为 L 的多项式 P
% 给定一个线性相位 FIR 滤波器的脉冲响应 h
% 此子程序能自动确定滤波器的类型
%
% Hr = 振幅响应
% w = 在 Hr 的计算区间[0,pi]上的频率
% P = 多项式系数
```

% L = P 的阶数

% h = 线性相位滤波器脉冲响应

它首先应确定线性相位 FIR 滤波器的类型,然后使用本章中相应的 *Hr_Type#* 函数。它也能判断给定的 $h(n)$是否是线性相位的。用例 7.4 至 7.7 中的序列验证你的子程序。

P7.6 如果 $H(z)$在

$$z_1 = re^{j\theta}, \quad z_2 = \frac{1}{r}e^{-j\theta}, \quad z_3 = re^{-j\theta}, \quad z_4 = \frac{1}{r}e^{-j\theta}$$

处有零点,证明 $H(z)$表示一个线性相位 FIR 滤波器。

P7.7 用汉宁窗设计技术设计一个带阻滤波器,技术指标为:

低阻带边缘: 0.4π

高阻带边缘: 0.6π $\quad A_s = 40$ dB

低通带边缘: 0.3π

高通带边缘: 0.7π $\quad R_p = 0.5$dB

画出设计的滤波器的脉冲响应和幅度响应(dB 值)。

P7.8 用哈明窗设计技术设计一个带通滤波器,技术指标为:

低阻带边缘: 0.3π

高阻带边缘: 0.6π $\quad A_s = 50$ dB

低通带边缘: 0.4π $\quad R_p = 0.5$dB

高通带边缘: 0.5π

画出设计的滤波器的脉冲响应和幅度响应(dB 值)。

P7.9 用凯泽窗设计技术设计一个高通滤波器,技术指标为:

阻带边缘: 0.4π, $\quad A_s = 60$dB

通带边缘: 0.6π, $\quad R_p = 0.5$dB

画出设计的滤波器的脉冲响应和幅度响应(dB 值)。

P7.10 用凯泽窗方法设计一个线性相位 FIR 数字滤波器,满足下面的技术指标:

$$\begin{aligned} 0 \leqslant |H(e^{j\omega})| \leqslant 0.01, &\quad 0 \leqslant \omega \leqslant 0.25\pi \\ 0.95 \leqslant |H(e^{j\omega})| \leqslant 1.05, &\quad 0.35\pi \leqslant \omega \leqslant 0.65\pi \\ 0 \leqslant |H(e^{j\omega})| \leqslant 0.01, &\quad 0.75\pi \leqslant \omega \leqslant \pi \end{aligned}$$

确定该滤波器脉冲响应的最小长度,给出它的振幅响应和幅度响应(dB 值)的图。

P7.11 参考本章用过的方法,开发下面的 MATLAB 函数,它们利用凯泽窗技术设计 FIR 滤波器。这些函数应能检验带的边缘频率是否正确,并限定滤波器的长度不超过 255。

a.低通滤波器:其格式为:

```
function[h,M] = kai_lpf(wp,ws,As);
%[h,M] = kai_lpf(wp,ws,As);
% 用凯泽窗设计低通 FIR 滤波器
%
% h = 所设计滤波器的脉冲响应,其长度为 M
% M = h 的长度,为奇数
% wp = 以弧度为单位的通带边缘频率(0 < wp < ws < pi)
% ws = 以弧度为单位的阻带边缘频率(0 < wp < ws < pi)
% As = 阻带衰减的 dB 数(As > 0)
```

b.高通滤波器:其格式为:

```
function[h,M] = kai_hpf(ws,wp,As);
%[h,M] = kai_hpf(ws,wp,As);
% 用凯泽窗设计高通 FIR 滤波器
%
% h = 所设计滤波器的脉冲响应,其长度为 M
% M = h 的长度,为奇数
% ws = 以弧度为单位的阻带边缘频率(0 < ws < wp < pi)
% wp = 以弧度为单位的通带边缘频率(0 < ws < wp < pi)
% As = 阻带衰减的 dB 数(As > 0)
```

c.带通滤波器:其格式为:

```
function[h,M] = kai_bpf(ws1,wp1,wp2,ws2,As);
%[h,M] = kai_bpf(ws1,wp1,wp1,ws2,As);
%用凯泽窗设计带通 FIR 滤波器
%
% h = 所设计滤波器的脉冲响应,其长度为 M
% M = h 的长度,为奇数
% ws1 = 以弧度为单位的低阻带边缘频率
% wp1 = 以弧度为单位的低通带边缘频率
% wp2 = 以弧度为单位的高通带边缘频率
% ws2 = 以弧度为单位的高阻带边缘频率
% 0 < ws1 < wp1 < wp2 < ws2 < pi
% As = 阻带衰减的 dB 数(As > 0)
```

d.带阻滤波器:其格式为:

```
function[h,M] = kai_bsf(wp1,ws1,ws2,wp2,As);
%[h,M] = kai_bsf(wp1,ws1,ws2,wp2,As);
%用凯泽窗设计带阻 FIR 滤波器
%
```

% h = 所设计滤波器的脉冲响应,其长度为 M
% M = h 的长度,为奇数
% wp1 = 以弧度为单位的低通带边缘频率
% ws1 = 以弧度为单位的低阻带边缘频率
% ws2 = 以弧度为单位的高阻带边缘频率
% wp2 = 以弧度为单位的高通带边缘频率
% 0 < wp1 < ws1 < ws2 < wp2 < pi
% As = 阻带衰减的 dB 数(As > 0)

对于本章讨论的其他窗技术可开发类似的函数。

P7.12 用布莱克曼窗技术设计例 7.26 中的阶梯滤波器,技术指标为:

第一段: $0 \leqslant \omega \leqslant 0.3\pi$　　理想增益 = 1,　　容限 $\delta_1 = 0.01$

第二段: $0.4\pi \leqslant \omega \leqslant 0.7\pi$　　理想增益 = 0.5,　　容限 $\delta_2 = 0.005$

第三段: $0.8\pi \leqslant \omega \leqslant \pi$　　理想增益 = 0,　　容限 $\delta_3 = 0.001$

比较这种滤波器与例 7.26 的滤波器的长度。并给出幅度响应图(*dB* 值)。

P7.13 设想一个截止频率 $\omega_c = 0.3\pi$ 的理想低通滤波器,我们选择用 40 个样本的频率采样结构逼近这个滤波器。

a.选择 ω_c 处的样本为 0.5,用朴素设计法计算 $h(n)$,并确定最小阻带衰减。

b.改变 ω_c 处的样本,确定它的最优值,以便得到最大的最小阻带衰减。

c.在一张图中画出上面两种设计的幅度响应(dB 值),并对结果进行说明。

P7.14 用频率采样法设计习题 7.7 中的带阻滤波器,选择适当的滤波器阶数,使得过渡带中有一个样本,设计过程中使用该样本的最优值。

P7.15 用频率采样法设计习题 7.8 中的带阻滤波器,适当选择滤波器阶数,使得过渡带中有两个样本,设计过程中使用这些样本的最优值。

P7.16 用频率采样法设计习题 7.9 中的高通滤波器,选择适当的滤波器阶数,使得过渡带中有两个样本,设计过程中使用这些样本的最优值。

P7.17 设计一个窄通带滤波器,通过的中心频率为 $\omega_0 = 0.5\pi$,其带宽应不大于 0.1π。

a. 利用频率采样技术,合理选择 M,使得过渡带中有一个样本,设计中使用过渡带样本的最优值,并画出频率采样结构。

b.利用凯泽窗技术,使阻带衰减与上面的频率采样设计中的结果相同,确定脉冲响应 $h(n)$,并画出线性相位结构。

c. 就上述两种滤波器设计的实现和滤波有效性进行比较。

P7.18 理想带通滤波器的频率响应为:

$$H_d(e^{j\omega}) = \begin{cases} 0, & 0 < |\omega| \leqslant \pi/3 \\ 1, & \pi/3 < |\omega| \leqslant 2\pi/3 \\ 0, & 2\pi/3 < |\omega| \leqslant \pi \end{cases}$$

a.根据 Parks-McClellan 算法,确定有 25 个抽头的滤波器的系数,它的阻带衰减为 50dB,设计的滤波器应该尽可能地做到具有最小的过渡带宽。

b.用在习题 7.5 中开发的函数画出滤波器的振幅响应。

P7.19 考察习题7.7中给出的带阻滤波器。

a. 用Parks-McClellan算法设计一个线性相位的带阻滤波器,滤波器的长度必须为奇数,画出所设计的滤波器的脉冲响应和幅度响应(dB)值。

b. 画出所设计的滤波器的振幅响应,并计算在阻带和通带中的极值总数,用理论估计的极值总数验证此结果。

c. 比较这个滤波器与习题7.7和7.14中的滤波器阶数。

d. 用下面的信号验证所设计的滤波器的功能。

$$x(n) = 5 - 5\cos\left(\frac{\pi n}{2}\right); 0 \leqslant n \leqslant 300$$

P7.20 用Parks-McClellan算法设计一个具有25个抽头的FIR微分器,它的斜率等于1个样本/周期。

a. 在0.1π至0.9π之间选择所关心的频率段,画出脉冲响应和振幅响应。

b. 产生100个正弦波信号样本

$$x(n) = 3\sin(0.25\pi n), n = 0, \ldots, 100$$

使样本通过上述的FIR微分器,将此结果与$x(n)$的理论求导相比较,不要忘了考虑FIR滤波器的12个样本的延迟。

P7.21 设计一个满足图7.48中的技术指标的最低阶等波动线性相位FIR滤波器,画出滤波器的脉冲响应和振幅响应。

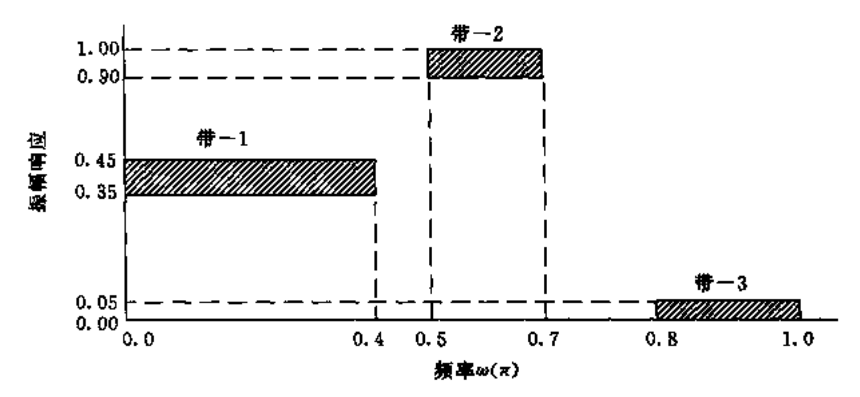


图7.48 习题7.21中的滤波器技术指标

P7.22 数字信号$x(n)$中包含频率为$\pi/2$的正弦波以及均值为零,方差为1的高斯噪声$\omega(n)$,即:

$$x(n) = 2\cos\frac{\pi n}{2} + \omega(n)$$

我们想用一个50阶因果的线性相位FIR滤波器滤除信号中的噪声分量。

a. 用Parks-McClellan算法设计一个窄带通滤波器,其通带宽度不大于0.02π,阻

带衰减至少为 30dB。注意没有给定其余的参数，为了满足要求，必须选择函数 remez 中所需的其他参数。画出所设计滤波器的对数幅度响应图。

b.产生 200 个序列 $x(n)$的样本，通过上述滤波器得到输出 $y(n)$，在同一张图上画出 $100 \leqslant n \leqslant 200$ 时的 $x(n)$和 $y(n)$，并对结果进行说明。

P7.23 设计一个满足下列技术指标的等波动数字希尔伯特变换器：

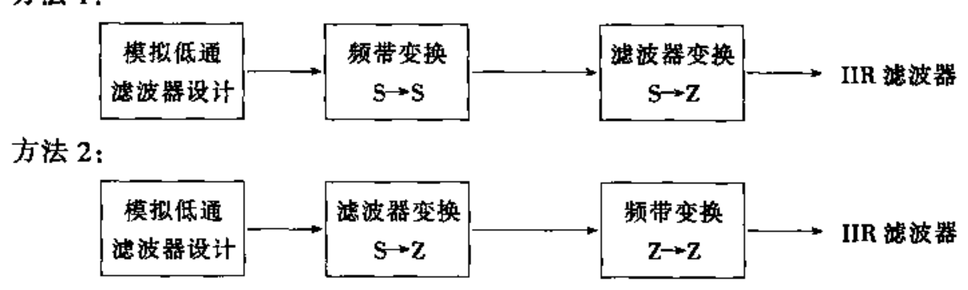
通带：$0.1\pi \leqslant |\omega| \leqslant 0.5\pi$ 容限 $\delta_1 = 0.01$

阻带：$0.5\pi \leqslant |\omega| \leqslant \pi$ 容限 $\delta_2 = 0.01$

画出 $-\pi \leqslant \omega \leqslant \pi$ 上的振幅响应。

第八章 IIR 滤波器设计

IIR 滤波器具有无限持续时间脉冲响应，所有的模拟滤波器一般都具有无限长的脉冲响应，因此它与模拟滤波器相匹配。IIR 滤波器设计的基本技术就是利用复值映射把熟知的模拟滤波器转换成数字滤波器。这种技术的优点是在许多著作中，模拟滤波器设计(AFD)表和映射均有现成的论述，它也叫做 A/D(模拟到数字)滤波器转换。但是，只有低通滤波器的 AFD 表是现成的。为了设计其他的选频滤波器(高通，带通，带阻等)，需要对低通滤波器进行频带变换，这些变换也是复值映射，许多文献中都论述过此映射。根据这种 IIR 滤波器设计的基本技术，引出两种设计方法。



在 MATLAB 中，采用第一种方法设计 IIR 滤波器。若直接运用这些 MATLAB 函数，则不能透彻地理解设计方法。因此，我们研究第二种方法，这是因为它涉及到数字域中的频带变换。在这种 FIR 滤波器设计技术中，遵循以下步骤：

·设计模拟低通滤波器。

·研究并进行滤波器变换，得到数字低通滤波器。

·研究并进行频带变换，从数字低通滤波器中得到其他类型的数字滤波器。

上述讨论的滤波器设计方法存在的主要问题是没有控制 IIR 滤波器的相位特性。因此把这种设计看作是只考虑幅度的设计。更成熟的技术可以同时逼近幅度和相位，它们需要更先进的优化工具箱，这本书里不涉及这些内容。

首先讨论模拟滤波器的技术指标，以及这些指标里用到的平方幅度响应特性。由此引出三种广泛应用的模拟滤波器，即巴特沃思、切比雪夫和椭圆滤波器。接着研究怎样把这些模拟滤波器原型转换成不同的选频数字滤波器。最后讨论和比较 FIR 和 IIR 滤波器的优缺点。

预 备 知 识

在本节中讨论两个问题，首先考虑平方幅度响应指标，它是模拟(因而也是 IIR)滤波器的典型指标，这些指标以相对线性标度的形式给出。接着，研究平方幅度响应的性质。

相对线性标度

设 $H_a(\Omega)$是一个滤波器的频率响应，则基于平方幅度响应的低通滤波器技术指标为：

$$\frac{1}{\sqrt{1+\epsilon^2}} \leqslant |H_a(j\Omega)|^2 \leqslant 1, \quad |\Omega| \leqslant \Omega_p$$
$$0 \leqslant |H_a(j\Omega)|^2 \leqslant \frac{1}{A^2}, \quad \Omega_s \leqslant |\Omega| \tag{8.1}$$

其中 ϵ 为通带波动系数，Ω_p 是单位为弧度/秒通带截止频率，A 为阻带衰减参数，Ω_s 是单位为弧度/秒的阻带截止频率。这些技术指标如图 8.1 所示。

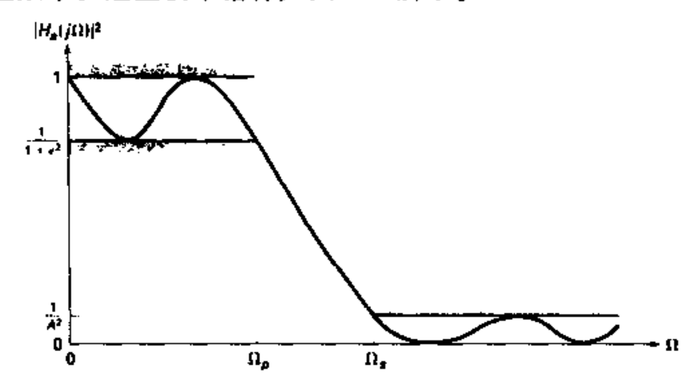


图 8.1 模拟低通滤波器技术指标

从图中可知 $|H_a(j\Omega)|^2$ 必须满足

$$|H_a(j\Omega_p)|^2 = \frac{1}{1+\epsilon^2} \qquad \Omega = \Omega_p \text{ 时}$$
$$|H_a(j\Omega)|^2 = \frac{1}{A^2} \qquad \Omega = \Omega_s \text{ 时} \tag{8.2}$$

参数 ϵ 和 A 以 dB 值的形式与 R_p 和 A_s 相关。其关系式为：

$$R_p = -10\log_{10}\frac{1}{1+\epsilon^2} \Rightarrow \epsilon = \sqrt{10^{R_p/10}-1} \tag{8.3}$$

$$A_s = -10\log_{10}\frac{1}{A^2} \Rightarrow A = 10^{A_s/20} \tag{8.4}$$

波动 δ_1、δ_2 以绝对量的形式与 ϵ 和 A 相关：

$$\frac{1-\delta_1}{1+\delta_1} = \sqrt{\frac{1}{1+\epsilon^2}} \Rightarrow \epsilon = \frac{2\sqrt{\delta_1}}{1-\delta_1}$$
$$\frac{\delta_2}{1+\delta_1} = \frac{1}{A} \Rightarrow A = \frac{1+\delta_1}{\delta_2}$$

$|H_a(j\Omega)|^2$ 的性质

模拟滤波器技术指标(8.1)以平方幅度响应的形式给出，它不包含相位信息。为了计算 s 域系统函数 $H_a(s)$，考虑到

$$H_a(j\Omega) = H_a(S)|_{s} = j\Omega$$

因此，有

$$|H_a(j\Omega)|^2 = H_a(j\Omega)H_a^*(j\Omega) = H_a(j\Omega)H_a(-j\Omega) = H_a(S)H_a(-S)|_{s=j\Omega}$$

或者

$$H_a(S)H_a(-S) = |H_a(j\Omega)|^2|_{\Omega=s/j} \tag{8.5}$$

所以，平方幅度函数的零、极点关于 $j\Omega$ 轴成镜像对称分布。对于实滤波器，零、极点以复共轭对(或关于实轴镜像对称)的形式出现。$H_a(s)H_a(-s)$的典型零、极点模式如图 8.2 所示。从中可以构造出模拟滤波器的系统函数 $H_a(s)$。我们希望 $H_a(s)$表示一个因果的、稳定的滤波器，则 $H_a(s)$所有的极点必须位于左半平面。我们把 $H_a(s)H_a(-s)$位于左半平面的极点分配给 $H_a(s)$。但 $H_a(s)$的零点可以位于 s 平面上的任意位置，因此除非它们在 $j\Omega$ 轴上，否则不能唯一确定，选择 $H_a(s)H_a(-s)$位于 $j\Omega$ 轴以左或在其上的零点作为 $H_a(s)$的零点。所得的滤波器叫做最小相位滤波器。

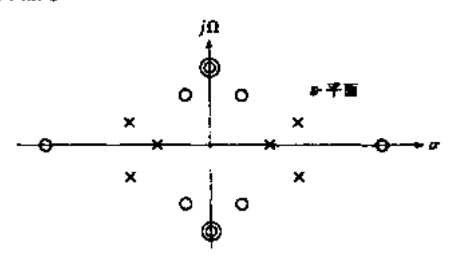


图 8.2 典型的 $H_a(s)H_a(-s)$零极点模式

模拟滤波器原型的特征

IIR 滤波器设计技术依靠现有的模拟滤波器得到数字滤波器，我们把这些模拟滤波器叫做滤波器原型。实际中有三种广泛应用的滤波器。在本节中，简要叙述这些低通滤波器原型的特征，即巴特沃思低通、切比雪夫低通以及椭圆低通滤波器。尽管我们用 MATLAB 函数设计这些滤波器，但是还是有必要知道这些滤波器的特征，以便在 MATLAB 函数中选择合适的参数得到正确的结果。

巴特沃思低通滤波器

这种滤波器的特征是其通带和阻带都有平的幅度响应。N 阶低通滤波器的平方幅度响

应为：

$$|H_a(j\Omega)|^2 = \frac{1}{1+\left(\frac{\Omega}{\Omega_c}\right)^{2N}} \tag{8.6}$$

其中，N 为滤波器的阶数，Ω_c 是截止频率(单位：弧度/秒)。平方幅度响应的图如下所示。

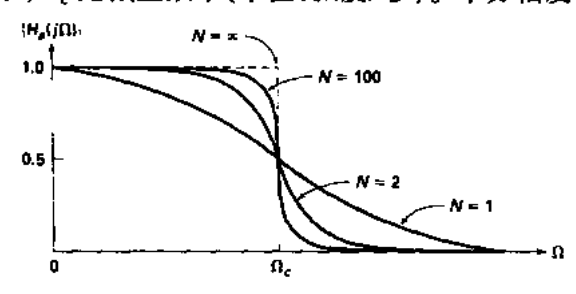


图 8.3 平方幅度响应图

从图中可以观察到下列性质：

·对于所有的 N，当 $\Omega=0$ 时，$|H_a(j0)|^2=1$。

·对于所有的 N，$\Omega=\Omega_c$ 时，$|H_a(j\Omega_c)|^2=1/2$，这说明在 Ω_c 处有 3dB 的衰减。

·$|H_a(j\Omega)|^2$ 是 Ω 的单调减函数。

·$N\to\infty$ 时 $|H_a(j\Omega)|^2$ 为趋于理想的低通滤波器。

·在 $\Omega=0$ 处各阶导数存在且等于 0，因此在这点上得到 $|H_a(j\Omega)|^2$ 的最大值。

为了确定系统函数 $H_a(s)$，把(8.6)代入(8.5)得到：

$$H_a(s)H_a(-s) = |H_a(j\Omega)|^2|_{\Omega=s/j} = \frac{1}{1+\left(\frac{s}{j\Omega_c}\right)^{2N}} = \frac{(j\Omega)^{2N}}{s^{2N}+(j\Omega_c)^{2N}} \tag{8.7}$$

从上式可得分母多项式的极点(或 $H_a(s)H_a(-s)$ 的根)为：

$$p_k = (-1)^{\frac{1}{2N}}(j\Omega) = \Omega_c e^{j\frac{\pi}{2N}(2k+N+1)},k=0,1,\ldots,2N-1 \tag{8.8}$$

对(8.8)的解释为：

·$H_a(s)H_a(-s)$ 有 $2N$ 个极点，等间隔分布在半径为 Ω_c 的圆上，其角度间隔为 π/N 弧度。

·N 为奇数时，极点 $p_k=\Omega_c e^{jk\pi/N}, k=0,1,\ldots,2N-1$,

·N 为偶数时，极点为 $p_k=\Omega_c e^{j\left(\frac{\pi}{2N}+\frac{k\pi}{N}\right)}, k=0,1,\ldots,2N-1$,

·极点关于 $j\Omega$ 对称。

·极点绝不会落在虚轴上，只有当 N 为奇数时会落在实轴上。

图 8.4 给出了三阶和四阶巴特沃思滤波器的极点分布图。

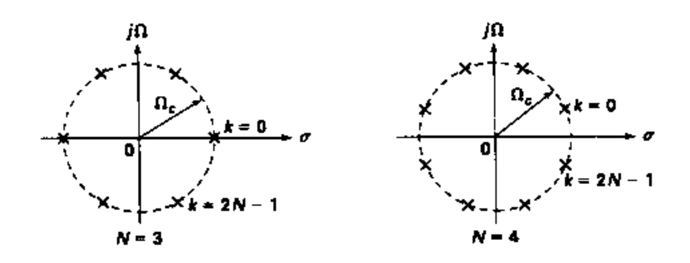


图 8.4　巴特沃思滤波器的极点分布图

通过选择左半平面的极点，得到一个稳定的、因果滤波器的 $H_a(s)$ 为：

$$H_a(s)=\frac{\Omega_c^N}{\prod\limits_{\text{LHP极点}}(s-p_k)} \tag{8.9}$$

例 8.1　给定 $|H_a(j\Omega)|^2=\frac{1}{1+64\Omega^6}$，确定模拟滤波器的系统函数 $H_a(s)$。

解：根据给定的平方幅度响应得：

$$|H_a(j\Omega)|^2=\frac{1}{1+64\Omega^6}=\frac{1}{1+\left(\frac{\Omega}{0.5}\right)^{2(3)}}$$

与(8.6)比较，得到 $N=3,\Omega_c=0.5$。$H_a(s)H_a(-s)$的极点如图 8.5 所示。

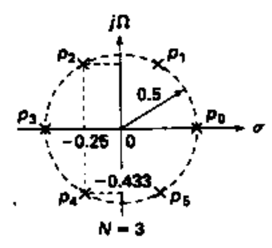


图 8.5　例 8.1 的极点图

因此

$$\begin{aligned}H_a(j\Omega)&=\frac{\Omega_c^3}{(s-s_2)(s-s_3)(s-s_4)}\\&=\frac{1/8}{(s+0.25-j0.433)(s+0.5)(s+0.25+j0.433)}\\&=\frac{0.125}{(s+0.5)(s^2+0.5s+0.25)}\end{aligned}$$

MATLAB 实现

MATLAB 提供了一个叫做[z,p,k] = buttap(N)的函数，来设计 N 阶归一化(即 $\Omega_c=1$)巴

特沃思模拟原型滤波器，它返回数组 z 和 p(零点和极点)以及增益 k。但是，我们需要具有任意 Ω_c 的未归一化的巴特沃思滤波器。例 8.1 的滤波器就没有零点，且极点位于半径为 Ω_c 的圆(而不是单位圆)上。这意味着需要用 Ω_c 对数组 p 进行归一化，用 Ω_c^N 对增益 k 进行归一化. 下面的 U_buttap(N, Omegac)函数，可用来设计未归一化的巴特沃思模拟原型滤波器。

```
function [b,a] = u_buttap(N,Omegac);
% 未归一化的巴特沃思模拟低通滤波器原型
% ---------------------------
% [b,a] = u_buttap(N,Omegac);
%   b = Ha(s) 分子多项式的系数
%   a = Ha(s) 分母多项式的系数
%   N = 滤波器的阶数
% Omegac = 单位为弧度/秒的截止频率
%
[z,p,k] = buttap(N);
          p = p * Omegac;
          k = k * Omegac^N;
          B = real(poly(z));
          b0 = k;
          b = k * B;
          a = real(poly(p));
```

上述函数提供了一个直接形式(或分子－分母)结构，用第六章中介绍的方法，可把直接形式转换成级联形式。下面的 sdir2cas 函数给出适合于模拟滤波器的做法。

```
function [C,B,A] = sdir2cas(b,a);
% s 平面中直接型到级联型的形式转换
% ---------------------------
% [C,B,A] = sdir2cas(b,a)
% C = 增益系数
% B = 包含各 bk 的 K 乘 3 维实系数矩阵
% A = 包含各 ak 的 K 乘 3 维实系数矩阵
% b = 直接型的分子多项式系数
% a = 直接型的分母多项式系数
%
Na = length(a) - 1; Nb = length(b) - 1;

% 计算增益系数 C
b0 = b(1); b = b/b0;
a0 = a(1); a = a/a0;
C = b0/a0;
%
%分母的二阶因子部分:
```

```
p= cplxpair(roots(a)); K = floor(Na/2);
if K*2 == Na      % 当 Na 为偶时计算
    A = zeros(K,3);
    for n=1:2:Na
        Arow = p(n:1:n+1,:);
        Arow = poly(Arow);
        A(fix((n+1)/2),:) = real(Arow);
    end

elseif Na == 1       % 当 Na = 1 时计算
    A = [0 real(poly(p))];

else                  % 当 Na 为奇及 > 1 时计算
    A = zeros(K+1,3);
    for n=1:2:2*K
        Arow = p(n:1:n+1,:);
        Arow = poly(Arow);
        A(fix((n+1)/2),:) = real(Arow);
        end
        A(K+1,:) = [0 real(poly(p(Na)))];
end

% 分子的二阶因子部分:
z = cplxpair(roots(b)); K = floor(Nb/2);
if Nb == 0            % 当 Nb = 0 时计算
    B = [0 0 poly(z)];

elseif K*2 == Nb      % 当 Nb 为偶时计算
    B = zeros(K,3);
    for n=1:2:Nb
        Brow = z(n:1:n+1,:);
        Brow = poly(Brow);
        B(fix((n+1)/2),:) = real(Brow);
    end

elseif Nb == 1        % 当 Nb = 1 时计算
    B = [0 real(poly(z))];

else                  % 当 Nb 为奇且 > 1 时计算
    B = zeros(K+1,3);
```

```
    for n = 1:2:2 * K
            Brow = z(n:1:n+1,:);
            Brow = poly(Brow);
            B(fix((n+1)/2),:) = real(Brow);
    end
    B(K+1,:) = [0 real(poly(z(Nb)))];
end
```

例 8.2 设计例8.1中给出的三阶巴特沃思模拟原型滤波器,其 $\Omega_c=0.5$。

解: MATLAB 程序

```
>>N = 3; OmegaC = 0.5;
>>[b,a] = u-buttap(N,OmegaC);
>>[C,B,A] = sdir2cas(b,a)
C = 0.1250
B = 0     0      1
A = 1.0000     0.5000     0.2500
         0     1.0000     0.5000
```

其级联系数与例 8.1 中的一致。

设计方程

模拟低通滤波器是由参数 Ω_p, R_p, Ω_s 和 A_s 描述的,设计巴特沃思滤波器的基本思想就是根据这些指标得到阶数 N 和截止频率 Ω_c。我们希望

· $\Omega=\Omega_p$ 时, $-10\log_{10}|H_a(j\Omega)|^2=R_p$

或

$$-10\log_{10}\left(\frac{1}{1+\left(\frac{\Omega_p}{\Omega_c}\right)^{2N}}\right)=R_p$$

并且

· $\Omega=\Omega_s$ 时, $-10\log_{10}|H_a(j\Omega)|^2=A_s$,

或

$$-10\log_{10}\left(\frac{1}{1+\left(\frac{\Omega_s}{\Omega_c}\right)^{2N}}\right)=A_s$$

将这两个方程对 N 和 Ω_c 求解,得:

$$N=\frac{\log_{10}[(10^{R_p/10}-1)(10^{A_s/10}-1)]}{2\log_{10}(\Omega_p/\Omega_s)}$$

一般情况下,上述的 N 不为整数。但 N 应该为整数,必须选择

$$N = \left\lceil \frac{\log_{10}[(10^{R_p/10} - 1)(10^{A_s/10} - 1)]}{2\log_{10}(\Omega_p/\Omega_s)} \right\rceil \tag{8.10}$$

运算$\lceil X \rceil$的意思是"选择比 x 大的最小整数",例如$\lceil 4.5 \rceil = 5$。由于选择的 N 比所需要的要大,在 Ω_p 或 Ω_s 处将满足或超过指标,为了在 Ω_p 处精确满足指标,

$$\Omega_c = \frac{\Omega_p}{\sqrt[2N]{(10^{R_p/10} - 1)}} \tag{8.11}$$

或在 Ω_s 处精确满足指标,

$$\Omega_c = \frac{\Omega_s}{\sqrt[2N]{(10^{A_s/10} - 1)}} \tag{8.12}$$

例 8.3 设计一个低通巴特沃思滤波器,以满足

通带截止频率:$\Omega_p = 0.2\pi$,　　通带波动:$R_p = 7\text{dB}$

阻带截止频率:$\Omega_s = 0.3\pi$,　　阻带波动:$A_s = 16\text{dB}$

解:根据(8.10)

$$N = \left\lceil \frac{\log_{10}[(10^{0.7} - 1)(10^{1.6} - 1)]}{2\log_{10}(0.2\pi/0.3\pi)} \right\rceil = \lceil 2.79 \rceil = 3$$

为了在 Ω_p 精确地满足指标,根据(8.11)得

$$\Omega_c = \frac{0.2\pi}{\sqrt[6]{(10^{0.7} - 1)}} = 0.4985$$

为了在 Ω_s 精确地满足指标,根据(8.12)得

$$\Omega_c = \frac{0.3\pi}{\sqrt[6]{(10^{1.6} - 1)}} = 0.5122$$

现在,可以在上面两个值之间选择任意的 Ω_c。因此选择 $\Omega_c = 0.5$.我们需要设计 $N = 3$ 和 $\Omega_c = 0.5$ 的巴特沃思滤波器,正如我们在例 8.1 中所做的。因此

$$H_a(j\Omega) = \frac{0.125}{(s + 0.5)(s^2 + 0.5s + 0.25)}$$

MATLAB 实现

上述设计过程在 MATLAB 中可以用一个简单函数来实现.利用 U _ buttap 函数,我们提供的 afd _ butt 函数可按给定的指标设计模拟巴特沃思低通滤波器,其中利用了(8.11)式。

```
function [b,a] = afd_butt(Wp,Ws,Rp,As);
% 巴特沃思型模拟低通滤波器设计
% -------------------------------
% [b,a] = afd_butt(Wp,Ws,Rp,As);
%  b = Ha(s) 分子的系数
%  a = Ha(s) 分母的系数
% Wp = 以弧度/秒为单位的通带边缘频率; Wp > 0
% Ws = 以弧度/秒为单位的阻带边缘频率; Ws > Wp > 0
% Rp = 通带中的振幅波动的 + dB 数; (Rp > 0)
% As = 阻带衰减的 + dB 数; (As > 0)
%
if Wp <= 0
        error('通带边缘必须大于 0')
end
if Ws <= Wp
        error('阻带边缘必须大于通带边缘')
end
if (Rp <= 0) | (As < 0)
        error('通带波动或阻带衰减必须大于 0')
end

N = ceil((log10((10^(Rp/10)-1)/(10^(As/10)-1)))/(2*log10(Wp/Ws)));
fprintf('\n*** Butterworth Filter Order = %2.0f \n',N)
OmegaC = Wp/((10^(Rp/10)-1)^(1/(2*N)));
[b,a]=u_buttap(N,OmegaC);
```

为了显示模拟滤波器的频域图,我们提供了一个叫做 freqs _ m 的函数,它是 MATLAB 里提供的 freqs 函数的修正版本。这个函数以绝对量的形式和相对的 dB 值形式计算幅度响应和相位响应。这个函数与原先讨论的 freqz _ m 函数类似,它们之间的主要差别是在函数 freqs _ m中,其响应算至最大频率 Ω_{max}.

```
function [db,mag,pha,w] = freqs_m(b,a,wmax);
% s 域频率响应计算:改进版本
% -----------------------------
% [db,mag,pha,w] = freqs_m(b,a,wmax);
```

```
%  db = [0 to wmax]区间内的相对幅度
%  mag = [0 to wmax]区间内的绝对幅度
%  pha = [0 to wmax] 区间内的相位响应
%    w = 含[0 to wmax]区间内的 500 个频率样本的数组
%    b = Ha(s)的分子多项式系数
%    a = Ha(s)的分母多项式系数
% wmax = 希望获得响应的频率区间的最大频率
%
w = [0:1:500] * wmax/500;
H = freqs(b,a,w);
mag = abs(H);
db = 20 * log10((mag + eps)/max(mag));
pha = angle(H);
```

模拟滤波器的脉冲响应 $h_a(t)$ 由 MATLAB 的 impulse 函数计算所得。

例 8.4 用 MATLAB 设计例 8.3 中的模拟巴特沃思低通滤波器。

解:MATLAB 程序 -

```
>> Wp = 0.2 * pi; Ws = 0.3 * pi; Rp = 7; As = 16;
>> Ripple = 10 ^ ( - Rp/20); Attn = 10 ^ ( - As/20);
>> % 模拟滤波器设计:
>> [b,a] = afd_butt(Wp,Ws,Rp,As);
*** 巴特沃思滤波器阶次 = 3
>> % 计算二阶环节:
>> [C,B,A] = sdir2cas(b,a)
C = 0.1238
B = 0     0     1
A = 1.0000    0.4985    0.2485
         0    1.0000    0.4985
>> % 计算频率响应:
>> [db,mag,pha,w] = freqs_m(b,a,0.5 * pi);
>> % 计算脉冲响应:
>> [ha,x,t] = impulse(b,a);
```
- -

系统函数为:

$$H_a(s) = \frac{0.1238}{(s^2 + 0.4985s + 0.2485)(s + 0.4985)}$$

这个 $H_a(s)$ 与例 8.3 中的稍微有些不同，因为在那个例子中，选择 $\Omega_c=0.5$，而在 afd_butt函数中，Ω_c 的选择在于使 Ω_p 处满足技术指标。在图 8.6 中给出滤波器的图。

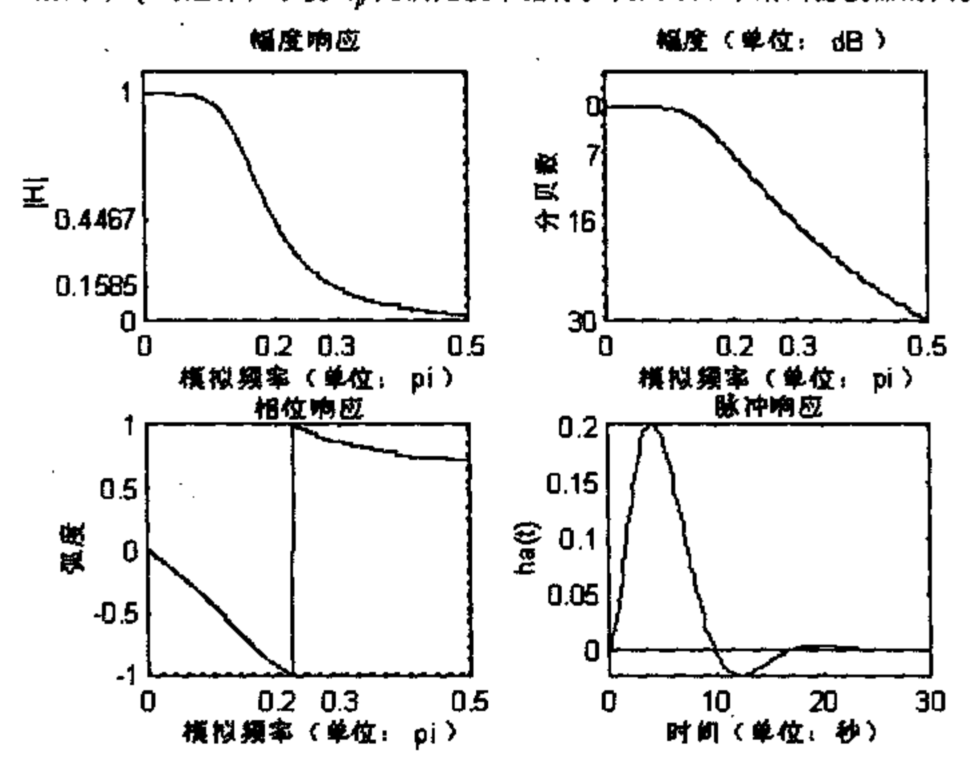


图 8.6 例 8.4 中的巴特沃思模拟滤波器

切比雪夫低通滤波器

切比雪夫滤波器有两种，切比雪夫Ⅰ型滤波器在通带中具有等波动响应，而切比雪夫Ⅱ型在阻带中具有等波动响应。巴特沃思滤波器在通带和阻带上的响应均是单调的。回忆讨论过的等波动 FIR 滤波器，可知选择等波动而不是具有单调特性的滤波器，可以得到较低阶的滤波器。因此，对于相同的指标，切比雪夫滤波器比巴特沃思滤波器的阶数低。

切比雪夫Ⅰ型滤波器的平方幅度响应为：

$$|H_a(j\Omega)|^2=\frac{1}{1+\epsilon^2T_N^2\left(\dfrac{\Omega}{\Omega_c}\right)} \tag{8.13}$$

其中 N 为滤波器的阶数，ϵ 为通带波动系数，它与 R_p 有关，$T_N(x)$ 是 N 阶切比雪夫多项式，

$$T_N(x)=\begin{cases}\cos(N\cos^{-1}(x)),0\leqslant x\leqslant 1\\ \cosh(\cosh^{-1}(x)),1<x<\infty\end{cases}\quad 其中\ x=\frac{\Omega}{\Omega_c}$$

切比雪夫滤波器的等波动响应是由多项式 $T_N(x)$ 造成的。它的主要性质是(a)$0<x<1$ 时，$T_N(x)$ 在 -1 至 1 之间出现振荡，(b)$0<x<\infty$ 时，$T_N(x)$ 单调增至 ∞。

$|H_a(j\Omega)|^2$ 有两种可能的形状，一种是 N 为奇数时，一种是 N 为偶数时，如图 8.7 所示。注意 $x=\Omega/\Omega_c$ 为归一化的频率。

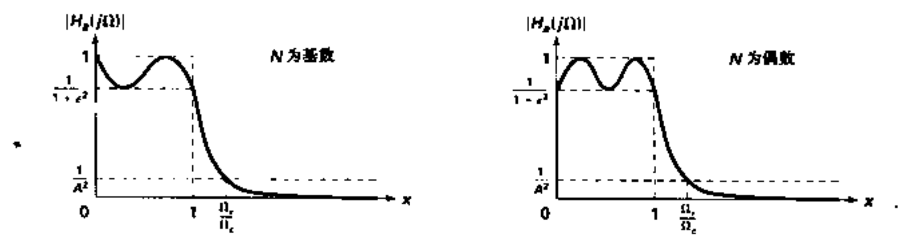


图 8.7　切比雪夫滤波器的 $|H_a(j\Omega)|^2$ 示意图

从上面两个图，观察到下列性质：

· $x=0$（或 $\Omega=0$）时，　$|H_a(j0)|^2=1$　　N 为奇数

$$|H_a(j0)|^2=\frac{1}{1+\epsilon^2}\qquad N\text{ 为偶数}$$

· $x=1$（或 $\Omega=\Omega_c$）时　$|H_a(j1)|^2=\dfrac{1}{1+\epsilon^2}$　　对所有的 N

· $0\leqslant x\leqslant 1$（或 $0\leqslant\Omega\leqslant\Omega_c$）时，$|H_a(jx)|^2$在 1 和$\dfrac{1}{1+\epsilon^2}$之间振荡

· $x>1$（或 $\Omega>\Omega_c$）时，$|H_a(jx)|^2$单调减至 0

· $x=\Omega_r$ 时，$|H_a(jx)|^2=\dfrac{1}{A^2}$

为了确定一个因果的和稳定的 $H_a(s)$，必须找出 $H_a(s)H_a(-s)$的极点，把左半平面的极点分配给 $H_a(s)$ 。通过求出

$$1+\epsilon^2 T_N^2\left(\frac{s}{j\Omega_c}\right)$$

的根得 $H_a(s)H_a(-s)$的极点。求解这个方程式即使不很困难，但也是很冗长的。可以证明如果 $p_k=\sigma_k+j\Omega_k$, $k=0,\ldots,N-1$ 是上述多项式的根（左半平面），则

$$\begin{aligned}\sigma_k&=(a\Omega_c)\cos\left[\frac{\pi}{2}+\frac{(2k+1)\pi}{2N}\right]\\ \Omega_k&=(b\Omega_c)\sin\left[\frac{\pi}{2}+\frac{(2k+1)\pi}{2N}\right]\end{aligned}\qquad k=0,\ldots,N-1 \tag{8.14}$$

其中

$$a=\frac{1}{2}\left(\sqrt[N]{\alpha}-\sqrt[N]{1/\alpha}\right),b=\frac{1}{2}\left(\sqrt[N]{\alpha}+\sqrt[N]{1/\alpha}\right),\text{以及},\alpha=\frac{1}{\epsilon}+\sqrt{1+\frac{1}{\epsilon^2}} \tag{8.15}$$

这些根落在长轴为 $b\Omega_c$，短轴为 $a\Omega_c$ 的椭圆上。此时系统函数为：

$$H_a(s)=\frac{K}{\prod_k(s-p_k)} \tag{8.16}$$

其中 K 是归一化因子,它使得

$$H_a(j0)=\begin{cases}1, & N\text{ 为奇数}\\ \dfrac{1}{\sqrt{1+\epsilon^2}}, & N\text{ 为偶数}\end{cases} \tag{8.17}$$

MATLAB 实现

MATLAB 提供了一个函数[z,p,k]=cheblap(N,Rp),来设计一个阶数为 N,通带波动为 Rp 的归一化切比雪夫Ⅰ型原型滤波器,它在数组 z 中返回零点,数组 p 中返回极点,并且返回增益 k。我们需要具有任意 Ω_c 的未归一化的切比雪夫Ⅰ型滤波器。这可由归一化滤波器的数组 p 乘以 Ω_c 得到。与巴特沃思原型类似,这种滤波器也没有零点。新增益 k 由(8.17)确定,即旧的增益 k 乘以非归一化与归一化多项式在 s=0 处的比值。在下面的函数 U_chb1ap (N,Rp,Omegac)中,设计一个未归一化的切比雪夫Ⅰ型模拟滤波器原型,它返回直接形式的 $H_a(s)$。

```
function [b,a] = u_chb1ap(N,Rp,Omegac);
% 未归一化的切比雪夫Ⅰ型模拟低通滤波器原型
% ---------------------------------------------------------------
% [b,a] = u_chb1ap(N,Rp,Omegac);
%    b = Ha(s) 分子多项式的系数
%    a = Ha(s) 分母多项式的系数
%    N = 滤波器的阶数
%    Rp = 通带波动 dB 数; Rp > 0
% Omegac = 以弧度/秒为单位的截止频率
%
[z,p,k] = cheb1ap(N,Rp);
         a = real(poly(p));
      aNn = a(N+1);
            p = p * Omegac;
            a = real(poly(p));
      aNu = a(N+1);
            k = k * aNu/aNn;
            b0 = k;
            B = real(poly(z));
            b = k * B;
```

设计方程

给定 Ω_p,R_p,Ω_s 和 A_s,为了求出切比雪夫Ⅰ型滤波器,需要知道三个参数 ϵ, Ω_c 和 N。根据式(8.3)和(8.4)得到

$\epsilon=\sqrt{10^{0.1R_p}-1}$ 及 $A=10^{A_s/20}$

从上面讨论的性质,可知

$$\Omega_c=\Omega_p \quad 及 \quad \Omega_r=\frac{\Omega_s}{\Omega_p} \tag{8.18}$$

阶数 N 为

$$g=\sqrt{(A^2-1)/\epsilon^2} \tag{8.19}$$

$$N=\left\lceil \frac{\log_{10}\left[g+\sqrt{g^2-1}\right]}{\log_{10}\left[\Omega_r+\sqrt{\Omega_r^2-1}\right]} \right\rceil \tag{8.20}$$

根据(8.15)、(8.14)和(8.16),可求出 $H_a(s)$.

例 8.5 设计一个低通切比雪夫Ⅰ型滤波器,满足

通带截止频率:$\Omega_p=0.2\pi$, 通带波动: $R_p=1\text{dB}$

阻带截止频率:$\Omega_s=0.3\pi$, 阻带波动: $A_s=16\text{dB}$

解:首先计算需要的参数:

$$\epsilon=\sqrt{10^{0.1(1)}-1}=0.5088 \qquad A=10^{16/20}=6.3096$$

$$\Omega_c=\Omega_p=0.2\pi \qquad \Omega_r=\frac{0.3\pi}{0.2\pi}=1.5$$

$$g=\sqrt{(A^2-1)/\epsilon^2}=12.2429 \qquad N=4$$

现在可以求出 $H_a(s)$

$$\alpha=\frac{1}{\epsilon}+\sqrt{1+\frac{1}{\epsilon^2}}=4.1702$$

$$a=0.5(\sqrt[N]{\alpha}-\sqrt[N]{1/\alpha})=0.3646$$

$$b=0.5(\sqrt[N]{\alpha}+\sqrt[N]{1/\alpha})=1.0644$$

$H_a(s)$有 4 个极点。

$$p_{0,3}=(a\Omega_c)\cos\left[\frac{\pi}{2}+\frac{\pi}{8}\right]\pm(b\Omega_c)\sin\left[\frac{\pi}{2}+\frac{\pi}{8}\right]=-0.0877\pm j0.6179$$

$$p_{1,2}=(a\Omega_c)\cos\left[\frac{\pi}{2}+\frac{3\pi}{8}\right]\pm(b\Omega_c)\sin\left[\frac{\pi}{2}+\frac{3\pi}{8}\right]=-0.2117\pm j0.2559$$

因此

$$H_a(s) = \frac{K}{\prod_{k=0}^{3}(s - p_k)} = \frac{0.03829(= 0.89125 \times .1103 \times .3895)}{(s^2 + 0.1754s + 0.3895)(s^2 + 0.4234s + 0.1103)}$$

注意其分子部分，可得

$$H_a(j0) = \frac{1}{\sqrt{1 + \epsilon^2}} = 0.89125$$

MATLAB 实现

以函数 U_chblap 为基础，我们提供了一个叫做 afd_chb1 的函数，它根据给定的技术指标设计模拟切比雪夫Ⅰ型滤波器。下面给出此函数，它采用了例 8.5 中叙述的方法。

```
function [b,a] = afd_chb1(Wp,Ws,Rp,As);
% 切比雪夫Ⅰ型模拟低通滤波器设计
% ---------------------------
% [b,a] = afd_chb1(Wp,Ws,Rp,As);
%   b = Ha(s) 分子的系数
%   a = Ha(s) 分母的系数
% Wp = 以弧度/秒为单位的通带边缘频率; Wp > 0
% Ws = 以弧度/秒为单位的阻带边缘频率; Ws > Wp > 0
% Rp = 通带中的振幅波动的 + dB 数; (Rp > 0)
% As = 阻带衰减的 + dB 数; (As > 0)
%
if Wp <= 0
        error('通带边缘必须大于 0')
end
if Ws <= Wp
        error('阻带边缘必须大于通带边缘')
end
if (Rp <= 0) | (As < 0)
        error('通带波动或阻带衰减必须大于 0')
end

ep = sqrt(10^(Rp/10) - 1);
A = 10^(As/20);
```

```
OmegaC = Wp;
OmegaR = Ws/Wp;
g = sqrt(A * A - 1)/ep;
N = ceil(log10(g + sqrt(g * g - 1))/log10(OmegaR + sqrt(OmegaR * OmegaR - 1)));
fprintf(' \n* * * 切比雪夫Ⅰ型 滤波器阶次 = %2.0f \n',N)
[b,a] = u_chb1ap(N,Rp,OmegaC);
```

例8.6 用MATLAB设计例8.5中的切比雪夫Ⅰ型滤波器。

解:MATLAB程序

```
>> Wp = 0.2 * pi; Ws = 0.3 * pi; Rp = 1; As = 16;
>> Ripple = 10 ^ ( - Rp/20); Attn = 10 ^ ( - As/20);
>> % 模拟滤波器设计:
>> [b,a] = afd_chb1(Wp,Ws,Rp,As);
* * * 切比雪夫Ⅰ型 滤波器阶次 = 4
>> % 计算二阶环节部分:
>> [C,B,A] = sdir2cas(b,a)
C = 0.0383
B = 0    0    1
A = 1.0000    0.4233    0.1103
    1.0000    0.1753    0.3895
>> % 计算频率响应:
>> [db,mag,pha,w] = freqs_m(b,a,0.5 * pi);
>> % 计算脉冲响应:
>> [ha,x,t] = impulse(b,a);
```

一个4阶的切比雪夫Ⅰ型滤波器技术指标,它的系统函数为:

$$H_a(s) = \frac{0.0383}{(s^2 + 4233s + 0.1103)(s^2 + 0.1753s + 0.3895)}$$

在图8.8中给出滤波器的响应曲线。

经过简单的变换,切比雪夫Ⅱ型滤波器可转换成切比雪夫Ⅰ型滤波器。它有一个单调的通带和等波动的阻带,这就意味着这种滤波器在s平面上既有极点,又有零点。因此通带中的群延迟特征比切比雪夫Ⅰ型滤波器好(而且相位响应更线性)。如果把(8.13)中 $\epsilon^2 T_N^2(\Omega/\Omega_c)$ 代换为它的倒数,并且把变量 $x = \Omega/\Omega_c$ 也换为它自己的倒数,我们得到切比雪夫Ⅱ型的平方幅度响应为:

$$|H_a(j\Omega)|^2 = \frac{1}{1 + [\epsilon^2 T_N^2(\Omega_c/\Omega)]^{-1}} \tag{8.21}$$

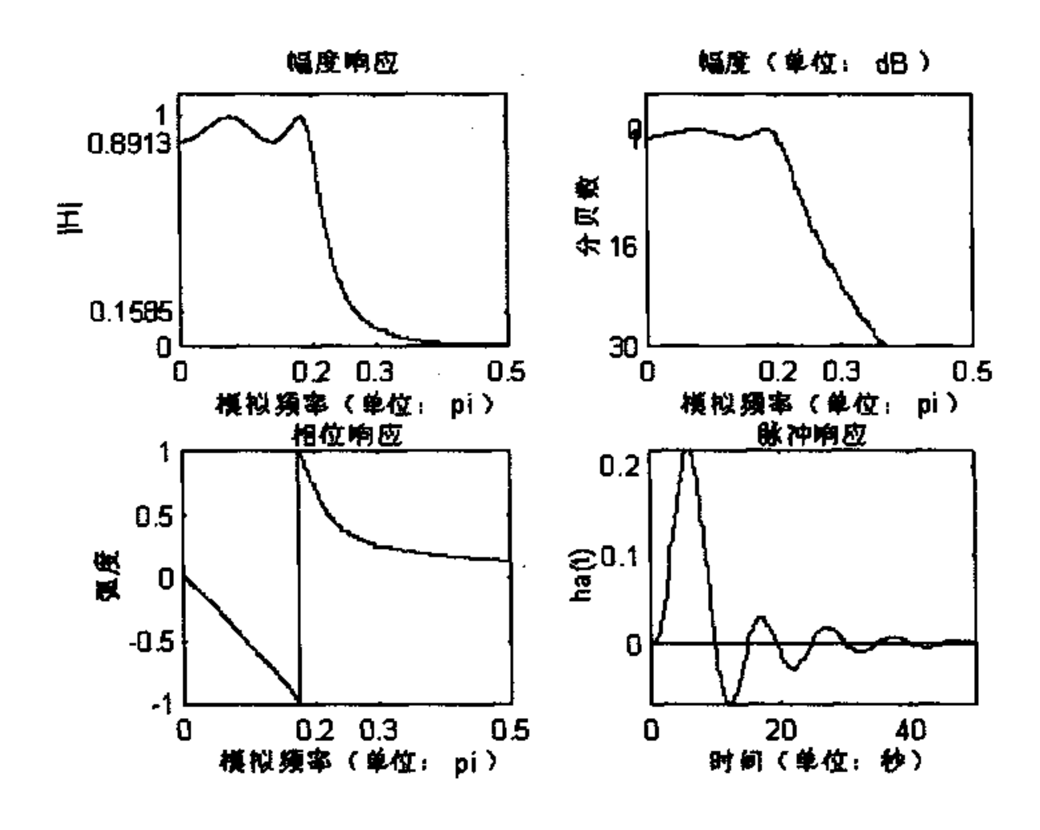


图 8.8　例 8.6 的切比雪夫Ⅰ型模拟滤波器

MATLAB 实现

MATLAB 提供了一个函数[z,p,k] = cheb2ap(N,As),用来设计阶数为 N,通带波动为 As 的归一化切比雪夫Ⅱ型原型滤波器,它在数组 z 中返回零点,数组 p 中返回极点,并且返回增益 k。我们需要具有任意 Ω_c 的未归一化的切比雪夫Ⅱ型滤波器,这可由归一化滤波器的数组 p 乘以 Ω_c 得到。同时由于它具有零点,数组 z 也必须乘以 Ω_c。新增益 k 由(8.17)确定,即旧的增益 k 乘以未归一化的多项式到归一化多项式的比在 s = 0 处的值。在下面的函数 U_chb2ap(N,As,Omegac)中,设计一个未归一化的切比雪夫Ⅱ型模拟滤波器原型,它返回直接形式的 $H_a(s)$。

```
function [b,a] = u_chb2ap(N,As,Omegac);
% 未归一化的切比雪夫Ⅱ型模拟低通滤波器原型
% -------------------------------
% [b,a] = u_chb2ap(N,As,Omegac);
%   b = Ha(s) 分子多项式的系数
%   a = Ha(s) 分母多项式的系数
%   N = 滤波器的阶数
%   As = 阻带波动 dB 数; As > 0
% Omegac = 以弧度/秒的截止频率
%
```

```
[z,p,k] = cheb2ap(N,As);
         a = real(poly(p));
       aNn = a(N+1);
           p = p * Omegac;
           a = real(poly(p));
       aNu = a(N+1);
           b = real(poly(z));
           M = length(b);
       bNn = b(M);
           z = z * Omegac;
           b = real(poly(z));
       bNu = b(M);
           k = k * (aNu * bNn)/(aNn * bNu);
         b0 = k;
             b = k * b;
```

除了 $\Omega_c = \Omega_s$ 以外，切比雪夫Ⅱ型原型的设计方程与切比雪夫Ⅰ型类似，这是因为波动是在阻带中。因此，我们可以为切比雪夫Ⅱ型原型开发一个与 afd_chb1 函数类似的 MATLAB 函数。

```
function [b,a] = afd_chb2(Wp,Ws,Rp,As);
* * * 切比雪夫Ⅱ型模拟低通滤波器设计
% ----------------------------------------
% [b,a] = afd_chb2(Wp,Ws,Rp,As);
%   b = Ha(s) 分子的系数
%   a = Ha(s) 分母的系数
% Wp = 以弧度/秒为单位的通带边缘频率; Wp > 0
% Ws = 以弧度/秒为单位的阻带边缘频率; Ws > Wp > 0
% Rp = 通带波动的 + dB 数; (Rp > 0)
% As = 阻带衰减的 + dB 数; (As > 0)
%
if Wp <= 0
        error('通带边缘必须大于 0')
end
if Ws <= Wp
      error('阻带边缘必须大于通带边缘')
end
if (Rp <= 0) | (As < 0)
      error('通带波动或阻带衰减必须大于 0')
end
```

```
ep = sqrt(10^(Rp/10) - 1);
A = 10^(As/20);
OmegaC = Wp;
OmegaR = Ws/Wp;
g = sqrt(A * A - 1)/ep;
N = ceil(log10(g + sqrt(g * g - 1))/log10(OmegaR + sqrt(OmegaR * OmegaR - 1)));
fprintf('\n*** 切比雪夫Ⅱ型 滤波器阶次 = %2.0f \n',N)
[b,a] = u_chb2ap(N,As,Ws);
```

例 8.7 设计一个切比雪夫Ⅱ型模拟低通滤波器满足例 8.5 中的技术指标：

通带截止频率：$\Omega_p = 0.2\pi$，　　通带波动：R_p = 1dB

阻带截止频率：$\Omega_s = 0.3\pi$，　　阻带波动：A_s = 16dB

解：MATLAB 程序

```
>> Wp = 0.2 * pi; Ws = 0.3 * pi; Rp = 1; As = 16;
>> Ripple = 10 ^ ( - Rp/20); Attn = 10 ^ ( - As/20);
>> % 模拟滤波器设计：
>> [b,a] = afd_chb2(Wp,Ws,Rp,As);
*** 切比雪夫Ⅱ型 滤波器阶次 = 4
>> % 计算二阶环节部分：
>> [C,B,A] = sdir2cas(b,a)
C = 0.1585
B = 1.0000         0    6.0654
    1.0000         0    1.0407
A = 1.0000    1.9521    1.4747
    1.0000    0.3719    0.6784

>> % 计算频率响应：
>> [db,mag,pha,w] = freqs_m(b,a,0.5 * pi);
>> % 计算脉冲响应：
>> [ha,x,t] = impulse(b,a);
```

一个四阶的切比雪夫Ⅱ型滤波器能够满足技术指标，它的系统函数为：

$$H_a(s) = \frac{0.1585(s^2 + 6.0654)(s^2 + 1.0407)}{(s^2 + 1.9521s + 1.4747)(s^2 + 0.3719s + 0.6784)}$$

在图 8.9 中给出滤波器的响应曲线。

椭圆低通滤波器

椭圆滤波器的通带和阻带均展现出等波动响应，它们的幅度响应特性与 FIR 等波动滤波

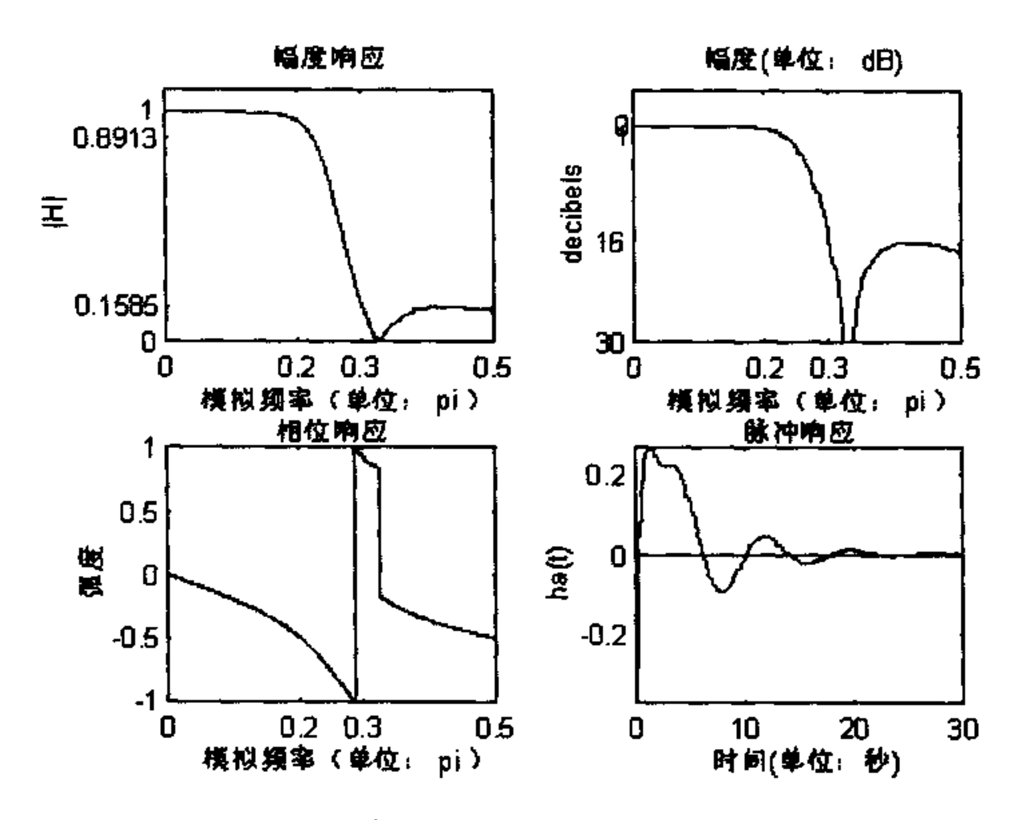


图 8.9　例 8.7 的切比雪夫Ⅱ型模拟滤波器

器类似，因此对于给定的指标，它们可使阶数 N 最小(换言之，给定阶数 N，使过渡带最陡)，从这个意义上讲，椭圆滤波器是最优滤波器。显然对这些滤波器很难进行分析，因而利用简单的工具不可能设计出它们，设计时经常需要现成的程序和表格。

椭圆滤波器的平方幅度响应为：

$$|H_a(j\Omega)|^2 = \frac{1}{1+\epsilon^2 U_N^2\left(\frac{\Omega}{\Omega_c}\right)} \tag{8.22}$$

其中 N 是阶数，ϵ 是通带波动(它与 R_p 有关)，$U_N(.)$是 N 阶雅可比椭圆函数，对此函数的分析，即使是很肤浅的，也超出了本书的范围。(8.22)所描述的响应与(8.13)给出的切比雪夫滤波器有相似之处。N 为奇数和偶数时的典型响应如图 8.10 所示。

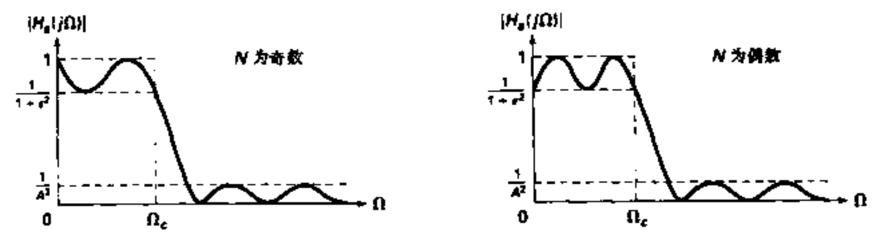


图 8.10　椭圆滤波器的平方幅度响应图

滤波器阶数 N 的计算

尽管很难对(8.22)进行分析,但是计算阶数的公式是很简洁的,这在许多教材中都可找到[文献 16,19,20],它的计算公式为

$$N = \frac{K(k)K(\sqrt{1-k_1^2})}{K(k_1)K(\sqrt{1-k^2})} \tag{8.23}$$

其中

$$k = \frac{\Omega_p}{\Omega_c}, k_1 = \frac{\epsilon}{\sqrt{A^2-1}}$$

并且

$$K(x) = \int_0^{\pi/2} \frac{d\theta}{\sqrt{1-x^2\sin^2\theta}}$$

是完全的椭圆积分,MATLAB 提供了 ellipke 函数对上述积分作数值计算,这会在计算 N 和设计椭圆滤波器时用到。

MATLAB 实现

MATLAB 提供了一个函数[z,p,k] = ellipap(N,Rp,As),来设计一个阶数为 N,通带波动为 Rp,阻带衰减为 As 的归一化椭圆模拟原型滤波器,它在数组 z 中返回零点,数组 p 中返回极点,并且返回增益 k。我们需要具有任意 Ω_c 的未归一化的椭圆滤波器,这可由归一化滤波器的数组 p 和 z 乘以 Ω_c,增益 k 乘以未归一化的多项式到归一化多项式的比在 s = 0 处的值得到。在下面的函数 U_elipap(N,Rp,As,Omegac)中,我们设计一个未归一化的椭圆模拟原型滤波器,它返回直接形式的 $H_a(s)$。

```
function [b,a] = u_elipap(N,Rp,As,Omegac);
% 未归一化的椭圆型模拟低通滤波器原型
% ---------------------------
% [b,a] = u_elipap(N,Rp,As,Omegac);
%    b = Ha(s) 分子多项式的系数
%    a = Ha(s) 分母多项式的系数
%    N = 滤波器的阶数
%    Rp = 通带波动 dB 数; Rp > 0
%    As = 阻带波动 dB 数; As > 0
% Omegac = 以弧度/秒的截止频率
```

```
%
[z,p,k] = ellipap(N,Rp,As);
         a = real(poly(p));
      aNn = a(N+1);
            p = p*Omegac;
            a = real(poly(p));
      aNu = a(N+1);
            b = real(poly(z));
            M = length(b);
      bNn = b(M);
            z = z*Omegac;
            b = real(poly(z));
      bNu = b(M);
            k = k*(aNu*bNn)/(aNn*bNu);
            b0 = k;
            b = k*b;
```

以 U_elipap 函数为基础,我们提供了一个如下所示的 afd_elip 的函数,它在给定技术指标下,设计模拟椭圆低通滤波器,同时它利用了滤波器阶数计算公式(8.23)。

```
function [b,a] = afd_elip(Wp,Ws,Rp,As);
% 椭圆模拟低通滤波器设计
%-----------------------
% [b,a] = afd_elip(Wp,Ws,Rp,As);
%   b = Ha(s) 分子的系数
%   a = Ha(s) 分母的系数
% Wp = 以弧度/秒为单位的通带边缘频率; Wp > 0
% Ws = 以弧度/秒为单位的阻带边缘频率; Ws > Wp > 0
% Rp = 通带波动的+dB 数; (Rp > 0)
% As = 阻带衰减的+dB 数; (As > 0)
%
if Wp <= 0
      error('通带边缘必须大于 0')
end
if Ws <= Wp
      error('阻带边缘必须大于通带边缘')
end
if (Rp <= 0) | (As < 0)
      error('通带波动或阻带衰减必须大于 0')
end
```

```
ep = sqrt(10^(Rp/10) - 1);
A = 10^(As/20);
OmegaC = Wp;
k = Wp/Ws;
k1 = ep/sqrt(A * A - 1);
capk = ellipke([k.^2 1 - k.^2]); % 4.0 版本代码
capk1 = ellipke([(k1 .^2) 1 - (k1 .^2)]); % 4.0 版本代码
N = ceil(capk(1) * capk1(2)/(capk(2) * capk1(1)));
fprintf(' \n * * * 椭圆滤波器阶次 = %2.0f \n',N)
[b,a] = u_elipap(N,Rp,As,OmegaC);
```

例 8.8 设计一个模拟椭圆低通滤波器满足例 8.5 的技术指标：

$$\Omega_p = 0.2\pi, \qquad R_p = 1\ \text{dB}$$
$$\Omega_s = 0.3\pi, \qquad A_s = 16\ \text{dB}$$

解：MATLAB 程序

```
>> Wp = 0.2 * pi; Ws = 0.3 * pi; Rp = 1; As = 16;
>> Ripple = 10 ^ ( - Rp/20); Attn = 10 ^ ( - As/20);
>> % 模拟滤波器设计：
>> [b,a] = afd_elip(Wp,Ws,Rp,As);
* * * 椭圆型滤波器阶次 = 3
>> % 计算二阶环节部分：
>> [C,B,A] = sdir2cas(b,a)
C = 0.2740
B = 1.0000         0    0.6641
A = 1.0000    0.1696    0.4102
         0    1.0000    0.4435
>> % 计算频率响应：
>> [db,mag,pha,w] = freqs_m(b,a,0.5 * pi);
>> % 计算脉冲响应：
>> [ha,x,t] = impulse(b,a);
```

用三阶的椭圆滤波器能够满足技术指标，它的系统函数为：

$$H_a(s) = \frac{0.274(s^2 + 0.6641)}{(s^2 + 0.1696s + 0.4102)(s + 0.4435)}$$

滤波器响应曲线如图 8.11 所示。

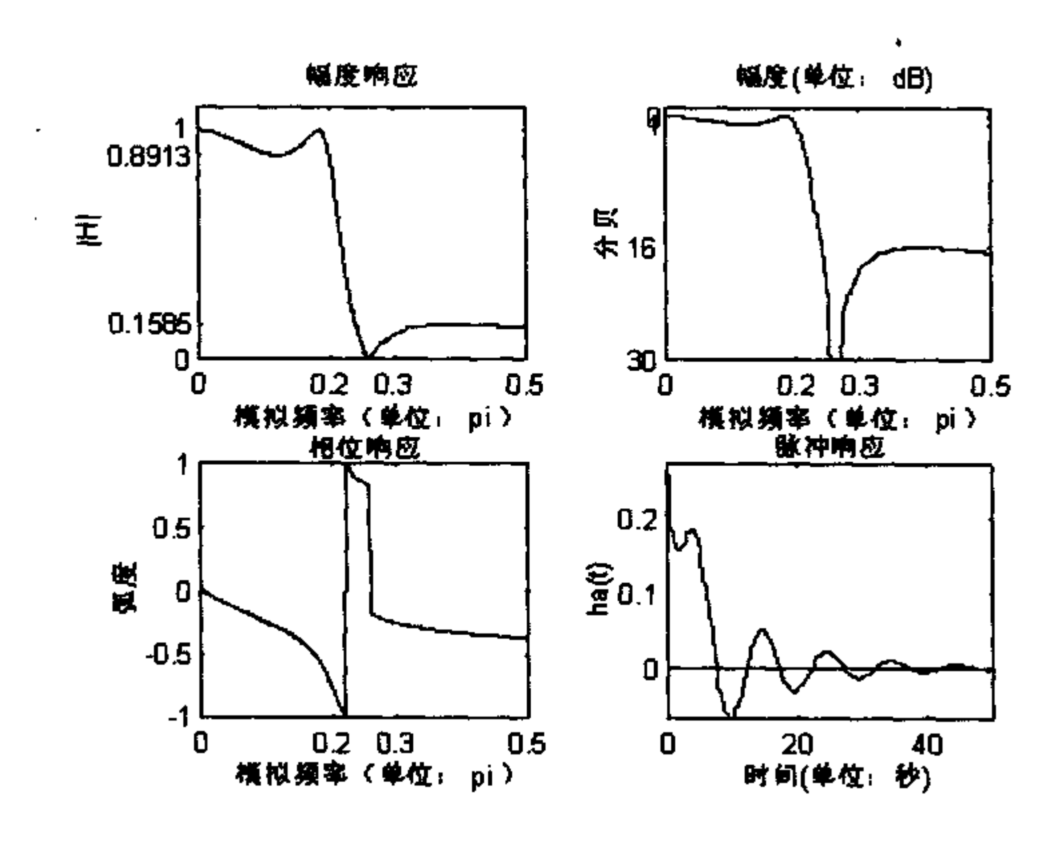


图 8.11 例 8.8 的椭圆模拟低通滤波器

滤波器原型的相位响应

椭圆滤波器提供了最优的平方幅度响应,但通带上的相位响应非线性较大(这是许多实际应用不希望的)。尽管在设计过程中,我们不考虑相位响应,但相位在整个系统中仍是个很重要的问题。性能尺度另一端是巴特沃思滤波器,它的幅度响应最单调,为了得到相同的阻带指标,它需要更高的阶数(更多的极点)。然而它在通带上具有相当线性的相位响应。切比雪夫滤波器的相位特征介于两者之间。因此,在实际应用中,我们既考虑巴特沃思和切比雪夫滤波器,同时也考虑椭圆滤波器。选择哪种类型则取决于滤波器的阶数(它影响处理的速度和实现的复杂性)和相位特性(它控制失真)。

模拟-数字滤波器变换

在讨论了模拟滤波器的各种设计方法后,我们就可以把它们变换成数字滤波器。这些变换均是复值映射,许多文献对此都有研究。根据数字滤波器所保持的模拟滤波器的不同特性,研究出不同的变换技术。如果希望保持脉冲响应的形状不变,则得到脉冲响应不变变换技术。如果希望把微分方程转换成相应的差分方程表示,则得到有限差分逼近技术。还可能有其他很多种技术,其中一种技术叫做阶跃不变,它保持阶跃响应的形状不变,这将在习题 9 中探讨。最受欢迎的一种技术是双线性变换,在从模拟到数字域变换过程中,它保持系统函数不变。本节将详细讨论脉冲响应不变和双线性变换技术,它们在 MATLAB 中均很容易实现。

脉冲响应不变变换

在这种设计方法中,希望数字滤波器的脉冲响应与模拟选频滤波器的脉冲响应相似。我们以采样间隔 T 对其进行采样,得到 $h(n)$;也就是,

$$h(n) = h_a(nT)$$

参数 T 的选择应使样本能"捕获"$h_a(t)$的形状。由于这是一个采样操作,则模拟和数字频率的关系为:

$$\omega = \Omega T \text{ 或 } e^{j\omega} = e^{j\Omega T}$$

由于在单位圆上 $z = e^{j\omega}$,在虚轴上 $s = j\Omega$,因此从 s 平面到 z 平面的变换为:

$$z = e^{sT} \tag{8.24}$$

根据频域混叠公式(3.27),系统函数 $H(z)$和 $H_a(s)$的关系为:

$$H(z) = \frac{1}{T}\sum_{k=-\infty}^{\infty} H_a\left(s - j\frac{2\pi}{T}k\right)$$

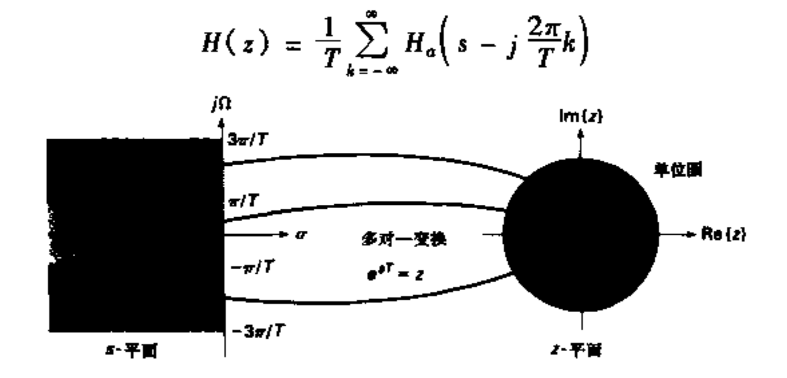


图 8.12　脉冲响应不变变换的复平面映射

经过(8.24)的映射,复平面变换如图 8.12 所示,从中可以看到:

1. 设$\sigma = \mathrm{Re}(s)$,则

   $\sigma < 0$　映射为$|z| < 1$(单位圆内)

   $\sigma = 0$　映射为$|z| = 1$(单位圆上)

   $\sigma > 0$　映射为$|z| > 1$(单位圆外)

2. 所有宽度为 $2\pi/T$ 的半无限带的映射均为$|z| < 1$,因此这个映射不是唯一的,而是多对一映射。

3. 由于 s 平面的左半平面映射为单位圆内,因此一个因果的和稳定的模拟滤波器映射成因果的和稳定的数字滤波器。

4. $|\Omega| \geqslant \pi/T$ 时,若 $H_a(j\Omega) = H_a(j\omega/T) = 0$,则

$$H(e^{j\omega}) = \frac{1}{T}H_a(j\omega/T), \quad |\omega| \leqslant \pi$$

不存在混叠。但是,有限阶数的模拟滤波器的带宽不可能是真正有限的,因此在这个设计过程中会出现混叠误差,而采样间隔 T 的影响则是次要的。

设计过程

给定数字低通滤波器技术指标 ω_p, ω_s, R_p, 和 A_s,我们首先设计一个等价的模拟滤波器,然后把它映射成所期望的数字滤波器。这个过程的设计步骤为:

1.确定 T 并选择模拟频率

$$\Omega_P = \frac{\omega_p}{T_p} \qquad \text{和} \qquad \Omega_s = \frac{\omega_s}{T}$$

2.根据指标 Ω_p, Ω_s, R_p, 和 A_s,设计模拟滤波器 $H_a(s)$。这可由上一节的三种原型(巴特沃思、切比雪夫、或椭圆)之一来实现。

3.利用部分分式展开式,把 $H_a(s)$展成

$$H_a(s) = \sum_{k=1}^{N} \frac{R_k}{s - p_k}$$

4.把模拟极点$\{p_k\}$转换成数字极点$\{e^{p_k T}\}$,得到数字滤波器:

$$H(z) = \sum_{k=1}^{N} \frac{R_k}{1 - e^{p_k T} z^{-1}} \tag{8.25}$$

例 8.9 利用脉冲响应不变技术,把

$$H_a(s) = \frac{s+1}{s^2 + 5s + 6}$$

转换成数字滤波器 $H(z)$,其中 $T = 0.1$.

解:首先,用部分分式展开 $H_a(s)$:

$$H_a(s) = \frac{s+1}{s^2 + 5s + 6} = \frac{2}{s+3} - \frac{1}{s+2}$$

极点为 $p_1 = -3$ 和 $p_2 = -2$,根据(8.25)和 $T = 0.1$,得:

$$H(z) = \frac{2}{1 - e^{-3T} z^{-1}} - \frac{1}{1 - e^{-2T} z^{-1}} = \frac{1 - 0.8966 z^{-1}}{1 - 1.5595 z^{-1} + 0.6065 z^{-2}}$$

很容易开发一个 MATLAB 函数实现脉冲响应不变映射。已知有理函数 $H_a(s)$,利用 residue 函数可以得到它的零极点。然后根据(8.24)把每一个模拟极点映射成数字极点。最后,函数

residuez 把 H(z)转换成有理函数形式。上述过程由函数 imp_invr 实现。

```
function [b,a] = imp_invr(c,d,T)
% 从模拟滤波器到数字滤波器的脉冲响应不变变换
% -----------------------------------------
% [b,a] = imp_invr(c,d,T)
%   b = 数字滤波器的自变量为 z^(-1)的分子多项式
%   a = 数字滤波器的自变量为 z^(-1)的分母多项式
%   c = 模拟滤波器的自变量为 s 的分子多项式
%   d = 模拟滤波器的自变量为 s 的分母多项式
%   T = 采样 (变换)参数
%
[R,p,k] = residue(c,d);
p = exp(p*T);
[b,a] = residuez(R,p,k);
b = real(b'); a = real(a');
```

在 MATLAB 新的学生版本中,提供了一个类似的函数 impinvar。

例 8.10 利用例8.9中的系统函数验证 imp_invr 函数。

解:MATLAB 程序——————————————

```
>> c = [1,1]; d = [1,5,6]; T = 0.1; Fs = 1/T;
  >> [b,a] = imp_invr(c,d,T)
  b = 1.0000   -0.8966
  a = 1.0000   -1.5595    0.6065
```

数字滤波器为:

$$H(z)=\frac{1-0.8966z^{-1}}{1-1.5595z^{-1}+0.6065z^{-2}}$$

与预想结果相同。在图 8.13 中给出模拟和所得数字滤波器脉冲响应和幅度响应(采样频率为 $1/T$)。显然,在频域中出现了混叠。

我们将在下面几个例子中,利用三种滤波器原型说明脉冲响应不变法的设计过程。

例 8.11 用巴特沃思滤波器原型设计一个低通数字滤波器,满足:

$$\omega_p=0.2\pi,\qquad R_p=1\text{dB}$$
$$\omega_s=0.3\pi,\qquad A_s=15\text{dB}$$

解:用下面的 MATLAB 程序说明设计过程。

```
>> % 数字滤波器指标:
>> wp = 0.2*pi;                        % 数字通带频率(弧度)
```

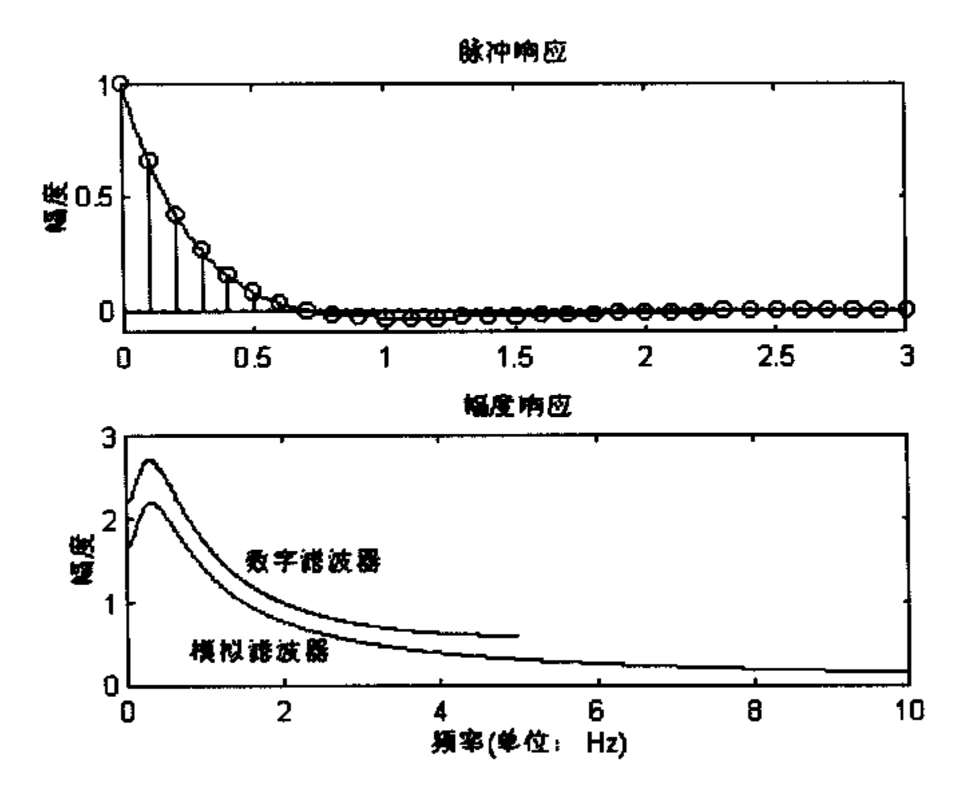


图 8.13 例 8.10 的脉冲和频率响应图

```
>> ws = 0.3 * pi;                                % 数字阻带频率(弧度)
>> Rp = 1;                                       % 通带波动(dB)
>> As = 15;                                      % 阻带衰减(dB)

>> % 模拟原型指标对频率的逆映射:
>> T = 1;                                        % 置 T=1
>> OmegaP = wp / T;                              % 原型通带频率
>> OmegaS = ws / T;                              % 原型阻带频率

>> % 模拟巴特沃思原型滤波器计算:
>> [cs,ds] = afd_butt(OmegaP,OmegaS,Rp,As);
* * * 巴特沃思滤波器阶次 = 6
>> % 脉冲恒定变换:
>> [b,a] = imp_invr(cs,ds,T);
>> [C,B,A] = dir2par(b,a)
C = []
B =1.8557   -0.6304
        -2.1428    1.1454
        0.2871    -0.4466
A = 1.0000   -0.9973    0.2570
```

```
    1.0000    -1.0691    0.3699
    1.0000    -1.2972    0.6949
```

所期望的滤波器是一个六阶的巴特沃思滤波器，其系统函数 $H(z)$ 的并联形式为：

$$H(z)=\frac{1.8587-0.6304z^{-1}}{1-0.9973z^{-1}+0.257z^{-2}}+\frac{-2.1428+1.1454z^{-1}}{1-1.0691z^{-1}+0.3699z^{-2}}+\frac{0.2871-0.4463z^{-1}}{1-1.2972z^{-1}+0.6449z^{-2}}$$

在图 8.14 中给出频率响应图。

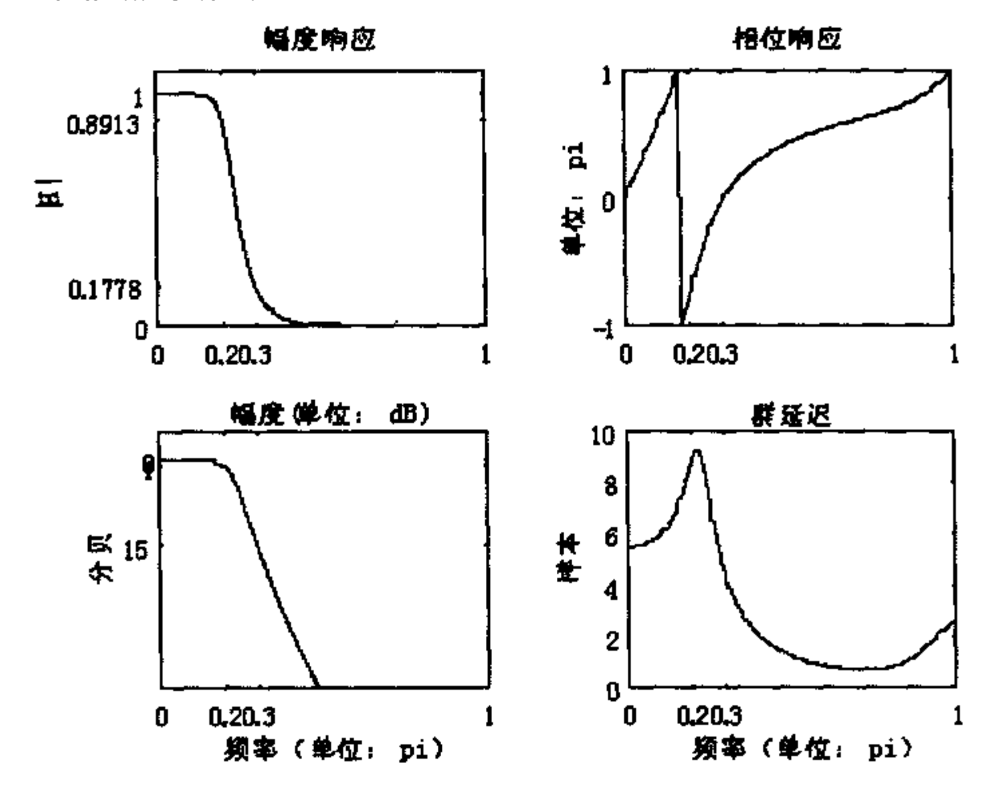


图 8.14　用脉冲响应不变法设计的数字巴特沃思低通滤波器

例 8.12　用切比雪夫 I 型滤波器原型设计一个低通数字滤波器，满足：

$$\omega_p=0.2\pi,\qquad R_p=1\text{dB}$$
$$\omega_s=0.3\pi,\qquad A_s=15\text{dB}$$

解：用下面的 MATLAB 程序说明设计过程：

```
>> % 数字滤波器指标：
>> wp = 0.2*pi;                    % 数字通带频率(弧度)
>> ws = 0.3*pi;                    % 数字阻带频率(弧度)
>> Rp = 1;                         % 通带波动(dB)
>> As = 15;                        % 阻带衰减(dB)
```

```
>> % 模拟原型指标的频率逆映射:
>> T = 1;                                % 置 T=1
>> OmegaP = wp / T;                      % 原型通带频率
>> OmegaS = ws / T;                      % 原型阻带频率

>> % 模拟切比雪夫Ⅰ型原型滤波器计算:
>>[cs,ds] = afd_chb1(OmegaP,OmegaS,Rp,As);
*** 切比雪夫Ⅰ型滤波器阶次 = 4

>> % 脉冲响应不变变换:
>> [b,a] = imp_invr(cs,ds,T);
>> [C,B,A] = dir2par(b,a)
C = []
B = -0.0833   -0.0246
       0.0833    0.0239
A = 1.0000   -1.4934    0.8392
      1.0000   -1.5658    0.6549
```

所期望的滤波器是一个四阶的切比雪夫Ⅰ型滤波器,其系统函数 $H(z)$为:

$$H(z)=\frac{-0.0833-0.0246z^{-1}}{1-1.4934z^{-1}+0.8392z^{-2}}+\frac{-0.0833+0.0239z^{-1}}{1-1.5658z^{-1}+0.6549z^{-2}}$$

在图 8.15 中给出频率响应图。

例 8.13 用切比雪夫Ⅱ型原型滤波器设计一个低通数字滤波器,满足:

$$\omega_p=0.2\pi,\qquad R_p=1\text{dB}$$
$$\omega_s=0.3\pi,\qquad A_s=15\text{dB}$$

解:切比雪夫Ⅱ型滤波器在阻带是等波动的,它在阻带中高频段的响应不趋于零。所以经过脉冲不变响应变换,其混叠效应会很明显;这会降低通带的响应。MATLAB 程序为:

```
>> % 数字滤波器指标:
>> wp = 0.2*pi;                          % 数字通带频率(弧度)
>> ws = 0.3*pi;                          % 数字阻带频率(弧度)
>> Rp = 1;                               % 通带波动(dB)
>> As = 15;                              % 阻带衰减(dB)

>> % 模拟原型指标的频率逆映射:
>> T = 1;                                % 置 T=1
>> OmegaP = wp / T;                      % 原型通带频率
>> OmegaS = ws / T;                      % 原型阻带频率
```

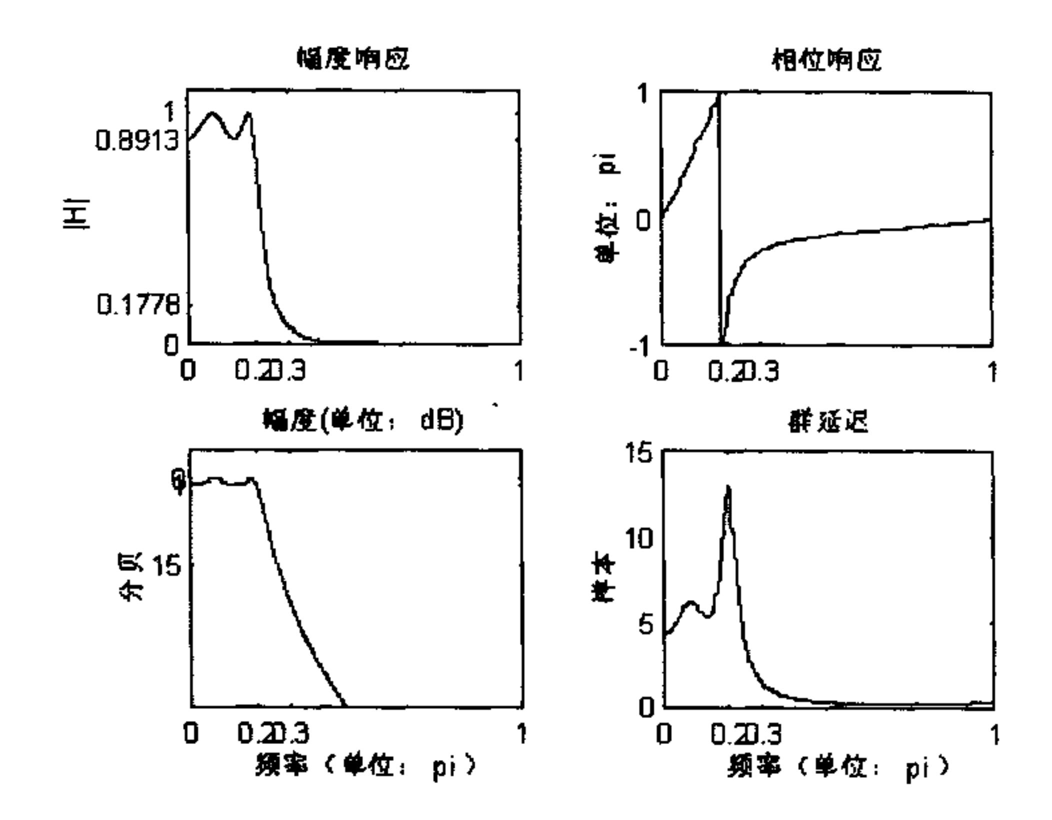


图 8.15　用脉冲响应不变法设计的数字切比雪夫Ⅰ型低通滤波器

```
>> % 模拟切比雪夫Ⅱ型原型滤波器计算:
>> [cs,ds] = afd_chb2(OmegaP,OmegaS,Rp,As);
* * * 切比雪夫Ⅱ型滤波器阶次 = 4

>> % 脉冲响应不变变换:
>> [b,a] = imp_invr(cs,ds,T);
>> [C,B,A] = dir2par(b,a)
```

由图 8.16 的频率响应图可以清楚地看到通带和阻带的特性变坏了。因此脉冲响应不变设计技术没能产生预期的数字滤波器。

例 8.14　用椭圆原型设计一个低通数字低通滤波器，满足

$$\omega_p = 0.2\pi, \qquad R_p = 1\text{dB}$$
$$\omega_s = 0.3\pi, \qquad A_s = 15\text{dB}$$

解：椭圆滤波器在通带和阻带上均是等波动的，因此这种情况与切比雪夫Ⅱ型滤波器相似，不会得到一个好的数字滤波器，MATLAB 程序为：

```
>> % 数字滤波器指标:
>> wp = 0.2 * pi;                          % 数字通带频率(弧度)
>> ws = 0.3 * pi;                          % 数字阻带频率(弧度)
>> Rp = 1;                                 % 通带波动(dB)
```

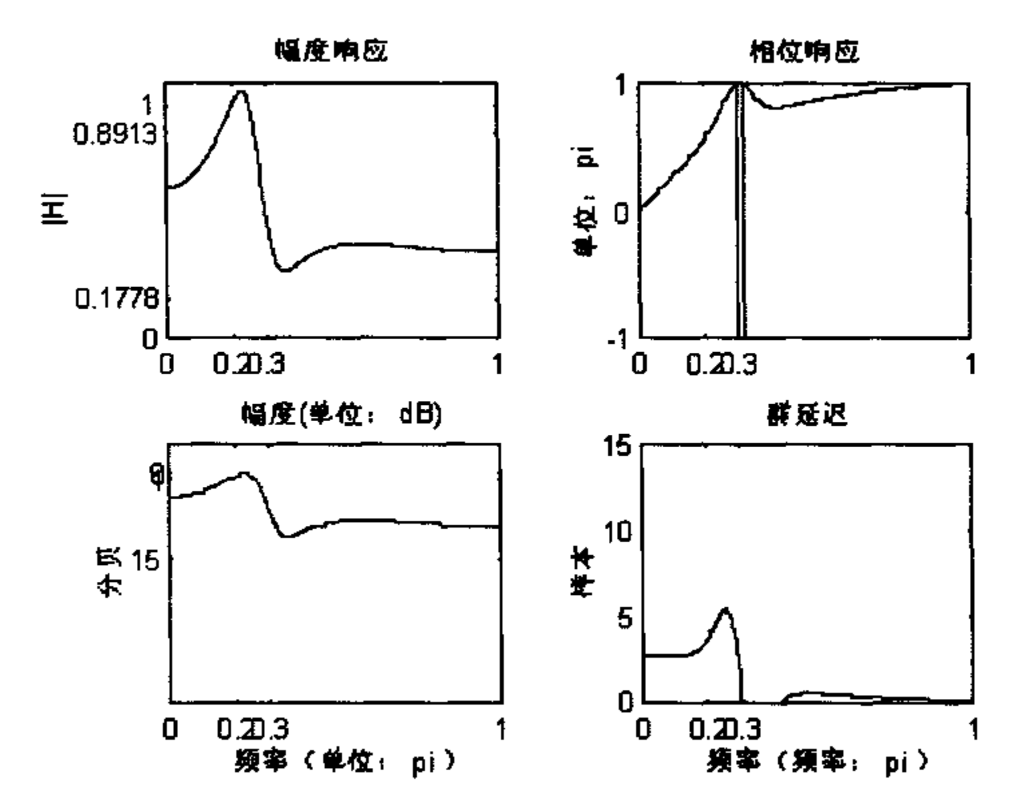


图 8.16　用脉冲响应不变法设计的数字切比雪夫Ⅱ型低通滤波器

```
> > As = 15;                                      % 阻带衰减(dB)

> > % 模拟原型指标的频率逆映射：
> > T = 1;                                        % 置 T = 1
> > OmegaP = wp / T;                              % 原型通带频率
> > OmegaS = ws / T;                              % 原型阻带频率

> > % 模拟椭圆原型滤波器计算：
> > [cs,ds] = afd_elip(OmegaP,OmegaS,Rp,As);
* * * 椭圆滤波器阶次 = 3

> > % 脉冲响应不变变换：
> > [b,a] = imp_invr(cs,ds,T);
> > [C,B,A] = dir2par(b,a)
```

从图 8.17 的频率响应图可以清楚地看到脉冲响应不变设计技术再次失败。

脉冲响应不变映射的优点在于它是一个稳定的设计，频率 Ω 和 ω 是线性相关的。但是它的缺点是模拟频率响应会有一些混叠，而且在某些情况下，这种混叠是不能忍受的。因此，只有当模拟滤波器基本上是有限带宽的低通或带通滤波器，而且在阻带中不存在振荡时，这种设计方法才是有用的。

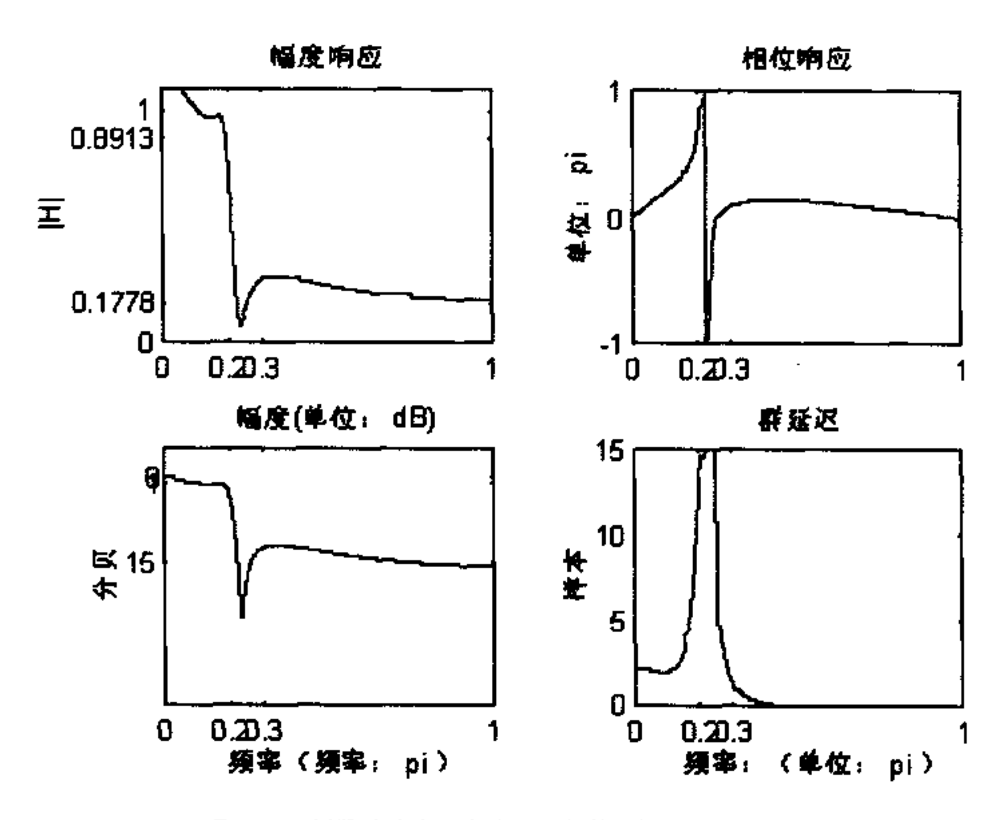


图 8.17　用脉冲响应不变法设计的数字椭圆低通滤波器

双线性变换

这种映射是最好的变换方法:它用到为人熟知的函数:

$$s=\frac{2}{T}\frac{1-z^{-1}}{1+z^{-1}}\Rightarrow z=\frac{1+sT/2}{1-sT/2} \tag{8.26}$$

其中 T 是参数,这种变换的另一个名字叫做线性分式变换,这是因为分式相除后,得到:

$$\frac{T}{2}sz+\frac{T}{2}s-z+1=0$$

若固定其中一个变量,则另一个是线性的。或者说,s 和 z 是双线性的。(8.26)的复平面映射如图 8.18 所示,从图中可以得出以下结论:

1.把 $s=\sigma+j\Omega$ 代入(8.26),得到

$$z=\left(1+\frac{\sigma T}{2}+j\frac{\Omega T}{2}\right)\Big/\left(1-\frac{\sigma T}{2}-j\frac{\Omega T}{2}\right) \tag{8.27}$$

因此

$$\sigma < 0 \Rightarrow |z| = \left| \frac{1 + \frac{\sigma T}{2} + j\frac{\Omega T}{2}}{1 - \frac{\sigma T}{2} - j\frac{\Omega T}{2}} \right| < 1$$

$$\sigma = 0 \Rightarrow |z| = \left| \frac{1 + j\frac{\Omega T}{2}}{1 - j\frac{\Omega T}{2}} \right| = 1$$

$$\sigma > 0 \Rightarrow |z| = \left| \frac{1 + \frac{\sigma T}{2} + j\frac{\Omega T}{2}}{1 - \frac{\sigma T}{2} - j\frac{\Omega T}{2}} \right| > 1$$

2.整个左半平面映射成单位圆的内部,因此这是一个稳定的变换。

3.虚轴以一一对应的方式映射成单位圆,因此在频域中不会出现混叠。

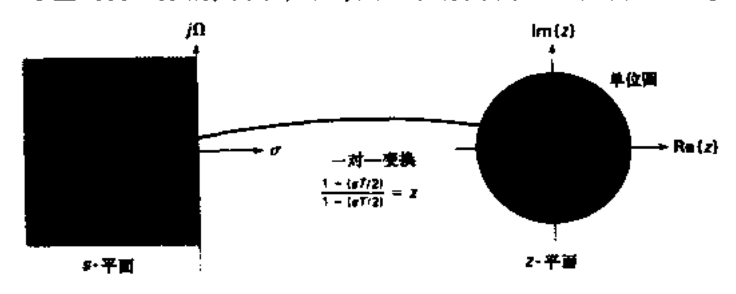


图 8.18　双线性变换的复平面映射

把 $\sigma = 0$ 代入(8.27),得到

$$z = \frac{1 + j\frac{\Omega T}{2}}{1 - j\frac{\Omega T}{2}} = e^{j\omega}$$

由于幅度为 1,把 ω 看作 Ω 的函数求解,得到

$$\omega = 2\tan^{-1}\left(\frac{\Omega T}{2}\right) \qquad \text{或 } \Omega = \frac{2}{T}\tan\left(\frac{\omega}{2}\right) \tag{8.28}$$

这说明 Ω 和 ω 的关系是非线性的(或 Ω 修正至 ω),但是没出现混叠。因此在(8.28)中,我们说 ω 预修正至 Ω。

例 8.15　利用双线性变换,把 $H_a(s) = \frac{s+1}{s^2+5s+6}$ 转换成数字滤波器,选择 $T = 1$。

解:根据(8.26),得到

$$H(z) = H_a\left(\frac{2}{T}\frac{1-z^{-1}}{1+z^{-1}}\bigg|_{T=1}\right) = H_a\left(2\frac{1-z^{-1}}{1+z^{-1}}\right)$$

$$= \frac{2\frac{1-z^{-1}}{1+z^{-1}}+1}{\left(2\frac{1-z^{-1}}{1+z^{-1}}\right)^2+5\left(2\frac{1-z^{-1}}{1+z^{-1}}\right)+6}$$

化简为：

$$H(z)=\frac{3+2z^{-1}-z^{-2}}{20+4z^{-1}}=\frac{0.15+0.1z^{-1}-0.05z^{-2}}{1+0.2z^{-1}}$$

MATLAB 提供了一个叫做 bilinear 的函数实现这种映射，它的性能与 imp_invr 函数相似，但它根据不同的输入输出数量，有几种不同的形式。为了了解更多的细节，可参考学生版本的手册。在下面的例子中将说明它的用途。

例 8.16 用 bilinear 函数对例 8.15 中的系统函数 $H_a(s)$ 进行变换。

解：MATLAB 程序

```
>> c = [1,1]; d = [1,5,6]; T = 1; Fs = 1/T
>> [b,a] = bilinear(c,d,Fs)
b = 0.1500    0.1000   -0.0500
a = 1.0000    0.2000    0.0000
```

滤波器为：

$$H(z)=\frac{0.15+0.1z^{-1}-0.05z^{-2}}{1+0.2z^{-1}}$$

与前面的相同。

设计过程

给定数字滤波器技术指标 ω_p，ω_s，R_p，和 A_s，要求确定 $H(z)$。设计步骤为：

1. 选择 T 值，它是任意的，也可以置为 1。

2. 对截止频率 ω_p 和 ω_s 进行预修正；也就是，根据(8.28)计算 Ω_p 和 Ω_s：

$$\Omega_p=\frac{2}{T}\tan\left(\frac{\omega_p}{2}\right),\ \Omega_s=\frac{2}{T}\tan\left(\frac{\omega_s}{2}\right) \tag{8.29}$$

3. 设计模拟滤波器 $H_a(s)$，使之满足技术指标 Ω_p，Ω_s，R_p，和 A_s，这在前一节中已讨论过。

4. 最后，设

$$H(z)=H_a\left(\frac{2}{T}\frac{1-z^{-1}}{1+z^{-1}}\right)$$

化简得到为 z^{-1}的有理函数的 $H(z)$。

在下面的几个例子中,我们用模拟滤波器原型说明此设计过程。

例 8.17 设计例8.11的数字巴特沃思滤波器。技术指标为:

$$\omega_p = 0.2\pi, \qquad R_p = 1\text{dB}$$
$$\omega_s = 0.3\pi, \qquad A_s = 15\text{dB}$$

解:MATLAB 程序

```
>> % 数字滤波器指标:
>> wp = 0.2 * pi;                          % 数字通带频率(弧度)
>> ws = 0.3 * pi;                          % 数字阻带频率(弧度)
>> Rp = 1;                                 % 通带波动(dB)
>> As = 15;                                % 阻带衰减(dB)

>> % 模拟原型指标的频率逆映射:
>> T = 1; Fs = 1/T;                        % 置 T=1
>> OmegaP = (2/T) * tan(wp/2);             % 预修正原型通带频率
>> OmegaS = (2/T) * tan(ws/2);             % 预修正原型阻带频率

>> % 模拟巴特沃思原型滤波器计算:
>> [cs,ds] = afd_butt(OmegaP,OmegaS,Rp,As);
*** 巴特沃思滤波器阶次 = 6

>> % 双线性变换:
>> [b,a] = bilinear(cs,ds,T);
>> [C,B,A] = dir2cas(b,a)
C = 5.7969e-004
B = 1.0000    2.0183    1.0186
    1.0000    1.9814    0.9817
    1.0000    2.0004    1.0000
A = 1.0000   -0.9459    0.2342
    1.0000   -1.0541    0.3753
    1.0000   -1.3143    0.7149
```

设计的滤波器也是一个六阶的滤波器,它有六个零点。由于 $H_a(s)$在 $s=-\infty$处的六重零点映射为 $z=-1$,这些零点应该在 $z=-1$ 处。由于 MATLAB 的精度有限,这些零点不会精确地位于 $z=-1$ 处,因此系统函数应该为:

$$H(z)=\frac{0.00057969(1+z^{-1})^6}{(1-0.9459z^{-1}+0.2342z^{-2})(1-1.0541z^{-1}+0.3753z^{-2})(1-1.3143z^{-1}+0.7149z^{-2})}$$

频率响应图如图 8.19 所示，可以看到这两种设计非常相似。

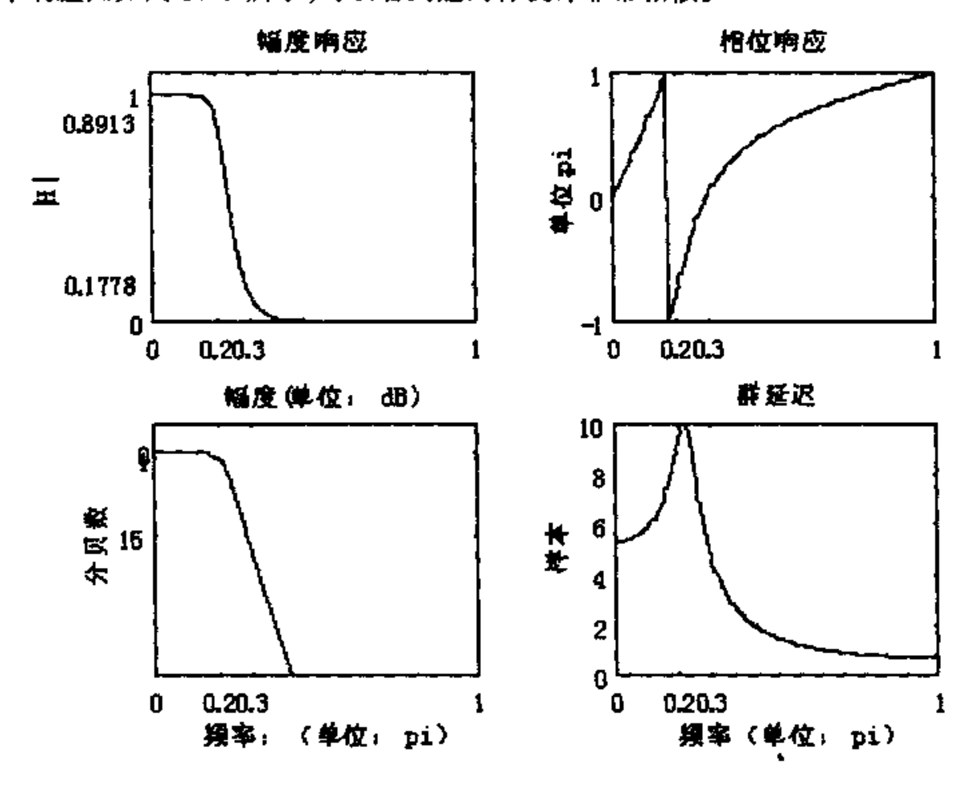


图 8.19 用双线性变换设计的数字巴特沃思低通滤波器

例 8.18 设计例8.12中的数字切比雪夫Ⅰ型滤波器，指标为：

$$\omega_p = 0.2\pi, \qquad R_p = 1\text{dB}$$
$$\omega_s = 0.3\pi, \qquad A_s = 15\text{dB}$$

解：MATLAB 程序——————————————

```
>> % 数字滤波器指标：
>> wp = 0.2*pi;                              % 数字通带频率(弧度)
>> ws = 0.3*pi;                              % 数字阻带频率(弧度)
>> Rp = 1;                                   % 通带波动(dB)
>> As = 15;                                  % 阻带衰减(dB)

>> % 模拟原型指标的频率逆映射：
>> T = 1; Fs = 1/T;                          % 置 T=1
>> OmegaP = (2/T)*tan(wp/2);                 % 预修正原型通带频率
>> OmegaS = (2/T)*tan(ws/2);                 % 预修正原型阻带频率

>> % 模拟切比雪夫Ⅰ型原型滤波器计算：
>> [cs,ds] = afd_chb1(OmegaP,OmegaS,Rp,As);
*** 切比雪夫Ⅰ型滤波器阶次 =   4
```

```
> > % 双线性变换：
> > [b,a] = bilinear(cs,ds,T);
> >[C,B,A] = dir2cas(b,a)
C = 0.0018
B = 1.0000    2.0000    1.0000
    1.0000    2.0000    1.0000
A = 1.0000   -1.4996    0.8482
    1.0000   -1.5548    0.6493
```

设计的滤波器是一个四阶滤波器，在 $z=-1$ 处有 4 重零点，系统函数为

$$H(z)=\frac{0.0018(1+z^{-1})^4}{(1-1.4996z^{-1}+0.8482z^{-2})(1-1.5548z^{-1}+0.6493z^{-2})}$$

频率响应图如图 8.20 所示，它与图 8.15 中的相似。

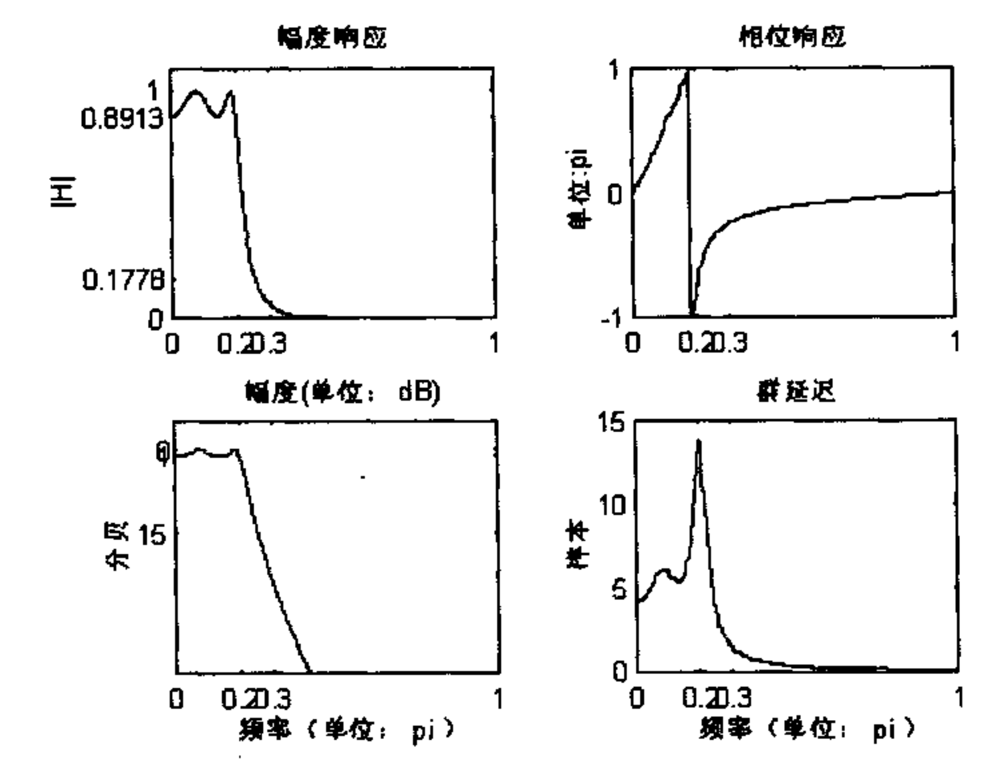


图 8.20　用双线性变换设计的数字切比雪夫 Ⅰ 型低通滤波器

例 8.19　设计例8.13中的数字切比雪夫 Ⅱ 型滤波器，技术指标为：

$$\omega_p=0.2\pi,\qquad R_p=1\text{dB}$$

$$\omega_s=0.3\pi,\qquad A_s=15\text{dB}$$

解：MATLAB 程序

```
> > % 数字滤波器指标：
> > wp = 0.2 * pi;                         % 数字通带频率(弧度)
```

```
>> ws = 0.3 * pi;                              % 数字阻带频率(弧度)
>> Rp = 1;                                     % 通带波动(dB)
>> As = 15;                                    % 阻带衰减(dB)

>> % 模拟原型指标的频率逆映射:
>> T = 1; Fs = 1/T;                            % 置 T=1
>> OmegaP = (2/T) * tan(wp/2);                 % 预修正原型通带频率
>> OmegaS = (2/T) * tan(ws/2);                 % 预修正原型阻带频率

>> % 模拟切比雪夫-Ⅱ原型滤波器计算:
>> [cs,ds] = afd_chb2(OmegaP,OmegaS,Rp,As);
*** 切比雪夫-Ⅱ滤波器阶次=  4

>> % 双线性变换:
>> [b,a] = bilinear(cs,ds,T);
>> [C,B,A] = dir2cas(b,a)
C = 0.1797
B = 1.0000    0.5574    1.0000
    1.0000   -1.0671    1.0000
A = 1.0000   -0.4183    0.1503
    1.0000   -1.1325    0.7183
```

设计的滤波器也是一个四阶滤波器,它的系统函数为

$$H(z)=\frac{0.1797(1+0.5574z^{-1}+z^{-2})(1-1.0671z^{-1}+z^{-2})}{(1-0.4183z^{-1}+0.1503z^{-2})(1-1.1325z^{-1}+0.7183z^{-2})}$$

频率响应图如图 8.21 所示,双线性变换较好地设计了切比雪夫Ⅱ型数字滤波器。

例 8.20 设计例8.14中的数字椭圆滤波器,技术指标为:

$$\omega_p=0.2\pi,\qquad R_p=1\text{dB}$$
$$\omega_s=0.3\pi,\qquad A_s=15\text{dB}$$

解:MATLAB 程序

```
>> % 数字滤波器指标:
>> wp = 0.2 * pi;                              % 数字通带频率(弧度)
>> ws = 0.3 * pi;                              % 数字阻带频率(弧度)
>> Rp = 1;                                     % 通带波动(dB)
>> As = 15;                                    % 阻带衰减(dB)

>> % 模拟原型指标的频率逆映射
```

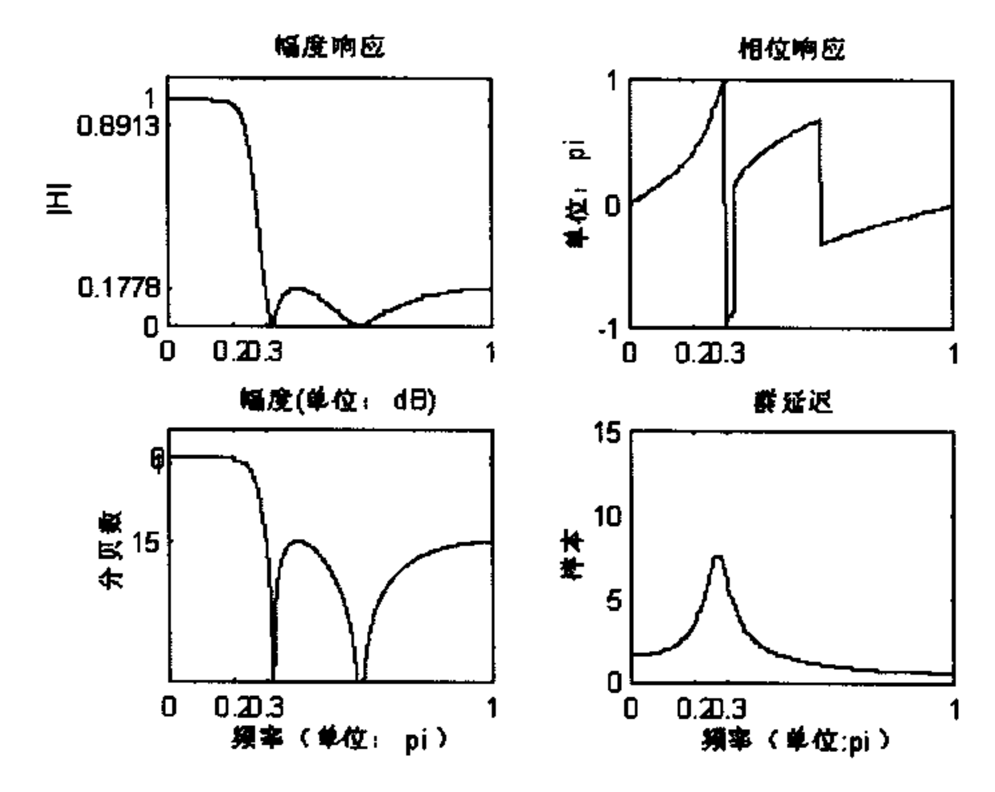


图 8.21 用双线性变换设计的数字切比雪夫Ⅱ型低通滤波器

```
>> T = 1; Fs = 1/T;                          % 置 T=1
>> OmegaP = (2/T) * tan(wp/2);               % 预修正原型通带频率
>> OmegaS = (2/T) * tan(ws/2);               % 预修正原型阻带频率

>> % 模拟椭圆原型滤波器计算:
>> [cs,ds] = afd_elip(OmegaP,OmegaS,Rp,As);
* * * 椭圆滤波器阶次 =   3

>> % 双线性变换:
>> [b,a] = bilinear(cs,ds,T);
>> [C,B,A] = dir2cas(b,a)
C = 0.1214
B = 1.0000   -1.4211    1.0000
    1.0000    1.0000         0
A = 1.0000   -1.4928    0.8612
    1.0000   -0.6183         0
```

所设计的滤波器是一个三阶的滤波器,其系统函数为:

$$H(z)=\frac{0.1214(1-1.4211z^{-1}+z^{-2})(1+z^{-1})}{(1-1.4928z^{-1}+0.8612z^{-2})(1-0.6183z^{-1})}$$

频率响应图如图 8.22 所示，双线性变换也较好地设计了椭圆数字滤波器。

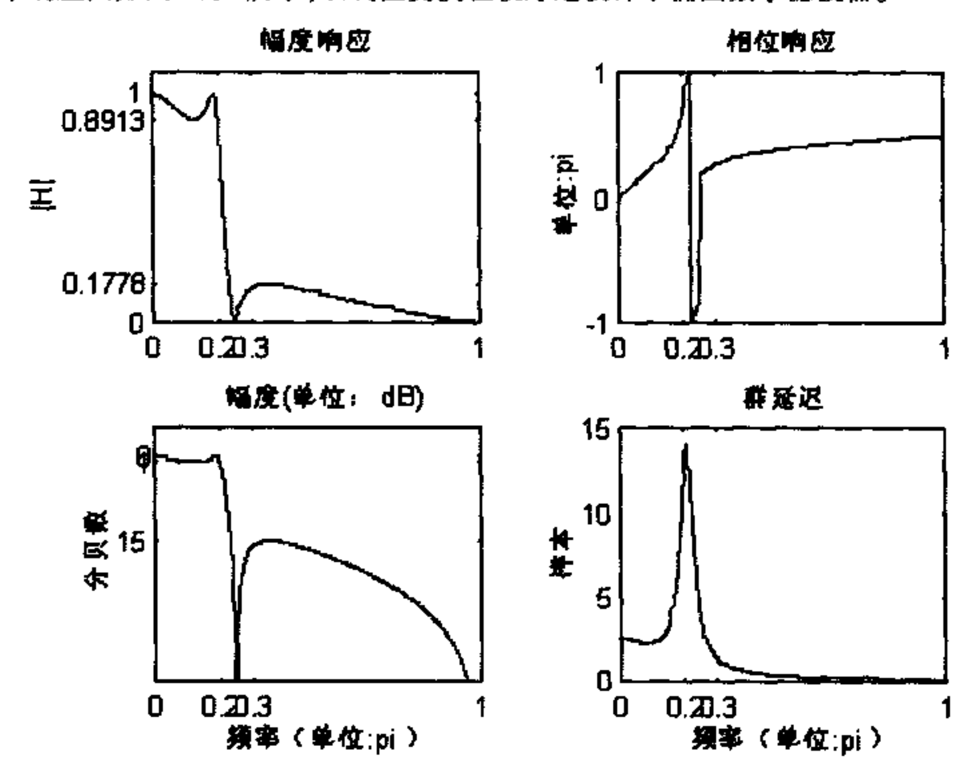


图 8.22 用双线性变换设计的数字椭圆低通滤波器

这种映射的优点是：(a)它是一个稳定设计；(b)不存在混叠；(c)对进行变换的滤波器类型没有限制，因此，这种方法广泛用在包括 MATLAB 在内的计算机程序中。

用 MATLAB 进行低通滤波器设计

在这一节中，我们将用实例说明用 MATLAB 程序来设计数字低通滤波器的方法。这些函数采用双线性变换，在上一节已讨论了它的优点。它们是：

1. [b,a] = butter(N,wn)

此函数设计 N 阶低通数字巴特沃思滤波器，在长度为 $N+1$ 的矢量 b 和 a 中返回滤波器系数。滤波器阶数由(8.10)确定，截止频率 ω_n 由预修正公式(8.29)得到。但是，在 MATLAB 中所有数字频率的单位为 π，因此，采用下式计算 ω_n：

$$\omega_n = \frac{2}{\pi}\tan^{-1}\left(\frac{\Omega_c T}{2}\right)$$

此函数的用法在例 8.21 中说明。

2. [b,a] = cheby1(N,Rp,wn)

此函数设计 N 阶低通数字切比雪夫 Ⅰ 型滤波器，通带波动为 R_p dB。在长度为 $N+1$ 的矢量 b 和 a 中返回滤波器系数。滤波器阶数由(8.20)确定，截止频率 ω_n 是单位为 π 的数字通

带频率,也就是:

$$\omega_n = \omega_p/\pi$$

此函数的用法在例 8.22 中说明。

3. [b,a] = cheby2(N,As,wn)

此函数设计 N 阶低通数字切比雪夫Ⅱ型滤波器,阻带衰减为 A_s dB。在长度为 $N+1$ 的矢量 b 和 a 中返回滤波器系数。滤波器阶数由(8.20)确定,截止频率 ω_n 是单位为π 的数字阻带频率,也就是:

$$\omega_n = \omega_s/\pi$$

此函数的用法在例 8.23 中说明。

4. [b,a] = ellip(N,Rp,As,wn)

此函数设计 N 阶低通数字椭圆滤波器,通带波动为 R_p dB,阻带衰减为 A_s dB。在长度为 $N+1$ 的矢量 b 和 a 中返回滤波器系数。滤波器阶数由(8.23)确定,截止频率 ω_n 是单位为π 的数字通带频率,也就是:

$$\omega_n = \omega_p/\pi$$

此函数的用法在例 8.24 中说明。

上述这些函数也可用来设计其他类型的选频滤波器,比如高通和带通滤波器。我们将在 8.5 节中讨论它们的这些功能。

也有另外一套滤波器函数,叫做 buttord,cheb1ord,cheb2ord 和 ellipord 函数,它们在给定的指标下确定阶数 N 和滤波器截止频率 ω_n。这些函数在信号处理工具箱里有现成的,但在学生版本中却没有,因此在下面的例子中,我们用前面所学的公式确定这些参数。在下一节中将讨论滤波器阶数函数。

下面对前面例题中的低通滤波器进行再设计,并比较它们的结果。其技术指标为:

$$\omega_p = 0.2\pi, \qquad R_p = 1\text{dB}$$
$$\omega_s = 0.3\pi, \qquad A_s = 15\text{dB}$$

例 8.21 数字巴特沃思低通滤波器设计:

```
>> % 数字滤波器指标:
>> wp = 0.2 * pi;                     % 数字通带频率(弧度)
>> ws = 0.3 * pi;                     % 数字阻带频率(弧度)
>> Rp = 1;                            % 通带波动(dB)
>> As = 15;                           % 阻带衰减(dB)

>> % 模拟原型指标:
```

```
>> T = 1;                                    % 置 T=1
>> OmegaP = (2/T) * tan(wp/2);               % 预修正原型通带频率
>> OmegaS = (2/T) * tan(ws/2);               % 预修正原型阻带频率

>> %模拟原型阶次计算:
>> N = ceil((log10((10^(Rp/10) - 1)/(10^(As/10) - 1)))/(2 * log10(OmegaP/
OmegaS)));
>> fprintf('\n* * * 巴特沃思滤波器阶次 = %2.0f \n',N)
* * * 巴特沃思滤波器阶次 = 6
>> OmegaC = OmegaP/((10^(Rp/10) - 1)^(1/(2 * N))); %模拟原型带宽截止频率
>> wn = 2 * atan((OmegaC * T)/2);            %数字带宽截止频率

>> % 数字巴特沃思滤波器设计:
>> wn = wn/pi;                               %以 pi 为单位的数字巴特沃思截止频
率
>> [b,a] = butter(N,wn);
>>[b0,B,A] = dir2cas(b,a)
b0 = 5.7969e-004
B = 1.0000    2.0297    1.0300
       1.0000    1.9997    1.0000
       1.0000    1.9706    0.9709
A = 1.0000   -0.9459    0.2342
       1.0000   -1.0541    0.3753
       1.0000   -1.3143    0.7149
```

系统函数为

$$H(z)=\frac{0.00057969(1+z^{-1})^{6}}{(1-0.9459z^{-1}+0.2342z^{-2})(1-1.0541z^{-1}+0.3753z^{-2})(1-1.3143z^{-1}+0.7149z^{-2})}$$

与例 8.17 结果相同。频率响应图如图 8.19 所示。

例 8.22 数字切比雪夫Ⅰ型低通滤波器设计:

```
>> % 数字滤波器指标:
>> wp = 0.2 * pi;                            % 数字通带频率(弧度)
>> ws = 0.3 * pi;                            % 数字阻带频率(弧度)
>> Rp = 1;                                   % 通带波动(dB)
>> As = 15;                                  % 阻带衰减(dB)

>> % 模拟原型指标:
```

```
>> T = 1;                                          % 置 T=1
>> OmegaP = (2/T) * tan(wp/2);                     % 预修正原型通带频率
>> OmegaS = (2/T) * tan(ws/2);                     % 预修正原型阻带频率

>> %模拟原型阶次计算:
>> ep = sqrt(10^(Rp/10) - 1);                      % 通带波动因子
>> A = 10^(As/20);                                 % 阻带波动因子
>> OmegaC = OmegaP;                                % 模拟原型截止频率
>> OmegaR = OmegaS/OmegaP;                         % 模拟原型传输比
>> g = sqrt(A * A - 1)/ep;                         % 模拟原型中间计算
>> N = ceil(log10(g + sqrt(g * g - 1))/log10(OmegaR + sqrt(OmegaR * OmegaR -
1)));
>> fprintf('\n*** 切比雪夫Ⅰ型滤波器阶次 = %2.0f \n',N)
*** 切比雪夫Ⅰ型滤波器阶次 =   4

>> % 数字切比雪夫Ⅰ型滤波器设计:
>> wn = wn/pi;                                     % 以 pi 为单位的数字通带频率
>> [b,a]=cheby1(N,Rp,wn);
>> [b0,B,A] = dir2cas(b,a)
b0 = 0.0018
B = 1.0000    2.0000    1.0000
    1.0000    2.0000    1.0000
A = 1.0000   -1.4996    0.8482
    1.0000   -1.5548    0.6493
```

系统函数为

$$H(z)=\frac{0.0018(1+z^{-1})^4}{(1-1.4996z^{-1}+0.8482z^{-2})(1-1.5548z^{-1}+0.6493z^{-2})}$$

与例 8.18 结果相同。频率响应图如图 8.20 所示。

例 8.23 数字切比雪夫Ⅱ型低通滤波器设计:

```
>> % 数字滤波器指标:
>> wp = 0.2 * pi;                                  % 数字通带频率(弧度)
>> ws = 0.3 * pi;                                  % 数字阻带频率(弧度)
>> Rp = 1;                                         % 通带波动(dB)
>> As = 15;                                        % 阻带衰减(dB)

>> % 模拟原型指标:
```

```
>> T = 1;                                             % 置 T=1
>> OmegaP = (2/T) * tan(wp/2);                        % 预修正原型通带频率
>> OmegaS = (2/T) * tan(ws/2);                        % 预修正原型阻带频率

>> %模拟原型阶次计算:
>> ep = sqrt(10^(Rp/10) - 1);                         % 通带波动因子
>> A = 10^(As/20);                                    % 阻带波动因子
>> OmegaC = OmegaP;                                   % 模拟原型截止频率
>> OmegaR = OmegaS/OmegaP;                            % 模拟原型传输比
>> g = sqrt(A * A - 1)/ep;                            % 模拟原型中间计算
>> N = ceil(log10(g + sqrt(g * g - 1))/log10(OmegaR + sqrt(OmegaR * OmegaR - 1)));
>> fprintf('\n*** 切比雪夫Ⅱ型滤波器阶次 = %2.0f \n',N)
*** 切比雪夫Ⅱ型滤波器阶次 = 4

>> % 数字切比雪夫Ⅱ型滤波器设计:
>> wn = wn/pi;                                        % 以 pi 为单位的数字阻带频率
>> [b,a]=cheby2(N,As,wn);
>> [b0,B,A] = dir2cas(b,a)
b0 = 0.1797
B = 1.0000    0.5574    1.0000
    1.0000   -1.0671    1.0000
A = 1.0000   -0.4183    0.1503
    1.0000   -1.1325    0.7183
```

系统函数为

$$H(z)=\frac{0.1797(1+0.5574z^{-1}+z^{-2})(1-1.0671z^{-1}+z^{-2})}{(1-0.4183z^{-1}+0.1503z^{-2})(1-1.1325z^{-1}+0.7183z^{-2})}$$

与例 8.19 结果相同。频率响应图如图 8.21 所示。

例 8.24 数字椭圆低通滤波器设计:

```
>> % 数字滤波器指标:
>> wp = 0.2 * pi;                                     % 数字通带频率(弧度)
>> ws = 0.3 * pi;                                     % 数字阻带频率(弧度)
>> Rp = 1;                                            % 通带波动(dB)
>> As = 15;                                           % 阻带衰减(dB)

>> % 模拟原型指标:
```

```
>> T = 1;                                          % 置 T=1
>> OmegaP = (2/T) * tan(wp/2);                     % 预修正原型通带频率
>> OmegaS = (2/T) * tan(ws/2);                     % 预修正原型阻带频率

>> % 模拟椭圆原型滤波器阶次计算:
>> ep = sqrt(10^(Rp/10) - 1);                      % 通带波动因子
>> A = 10^(As/20);                                 % 阻带波动因子
>> OmegaC = OmegaP;                                % 模拟原型截止频率
>> k = OmegaP/OmegaS;                              % 模拟原型传输比
>> k1 = ep/sqrt(A * A - 1);                        % 模拟原型中间计算
>> capk = ellipke([k.^2 1 - k.^2]);
>> capk1 = ellipke([(k1 .^2) 1 - (k1 .^2)]);
>> N = ceil(capk(1) * capk1(2)/(capk(2) * capk1(1)));
>> fprintf('\n* * * 椭圆滤波器阶次 = %2.0f \n',N)
* * * 椭圆滤波器阶次 =  3

>> % 数字椭圆滤波器设计
>> wn = wp/pi;                                     % 以 pi 为单位的数字通带频率
>> [b,a] = ellip(N,Rp,As,wn);
>> [b0,B,A] = dir2cas(b,a)
b0 = 0.1214
B = 1.0000   -1.4211    1.0000
         1.0000    1.0000         0
A = 1.0000   -1.4928    0.8612
         1.0000   -0.6183         0
```

系统函数为

$$H(z)=\frac{0.1214(1-1.4211z^{-1}+z^{-2})(1+z^{-1})}{(1-1.4928z^{-1}+0.8612z^{-2})(1-0.6183z^{-1})}$$

与例 8.20 结果相同。频率响应图如图 8.22 所示。

三种滤波器的比较

在上面的例子中,我们用四种不同的模拟滤波器原型设计相同的数字滤波器。现在比较它们的性能。指标为:$\omega_p=0.2\pi$, $R_p=1\text{dB}$, $\omega_s=0.3\pi$, $A_s=15\text{dB}$。表 8.1 给出按阶数 N 和最小阻带衰减所作的比较。

表 8.1 三种滤波器的比较

原型	阶数	阻带衰减
巴特沃思	6	15
切比雪夫-Ⅰ	4	25
椭圆	3	27

显然，椭圆滤波器原型给出的设计最佳。但是，若比较它们的相位响应，则在通带上，椭圆滤波器设计具有最明显的非线性相位响应。

频带变换

在前两节中，用模拟滤波器设计出相应的数字低通滤波器。此方法也能推广到设计其他类型的选频滤波器，比如高通、带通以及带阻滤波器。通过对低通滤波器的频率轴(或频带)进行变换，可以得到其他类型的选频器。这些在复变量 z 上的变换与双线性变换很相似，设计方程也是代数方程。设计一般的选频滤波器的过程为，首先设计一个数字(固定带宽，比如单位带宽)低通滤波器原型，然后进行代数变换。在这一节中将讨论隐藏在这些映射后的基本原理，并且通过例子说明其设计技巧。MATLAB 提供了能在 s 平面上实现频带变换的函数。我们首先用例子说明 z 平面映射的用法，接着给出 MATLAB 函数的使用方法。图 8.23 给出常用选频数字滤波器的典型指标。

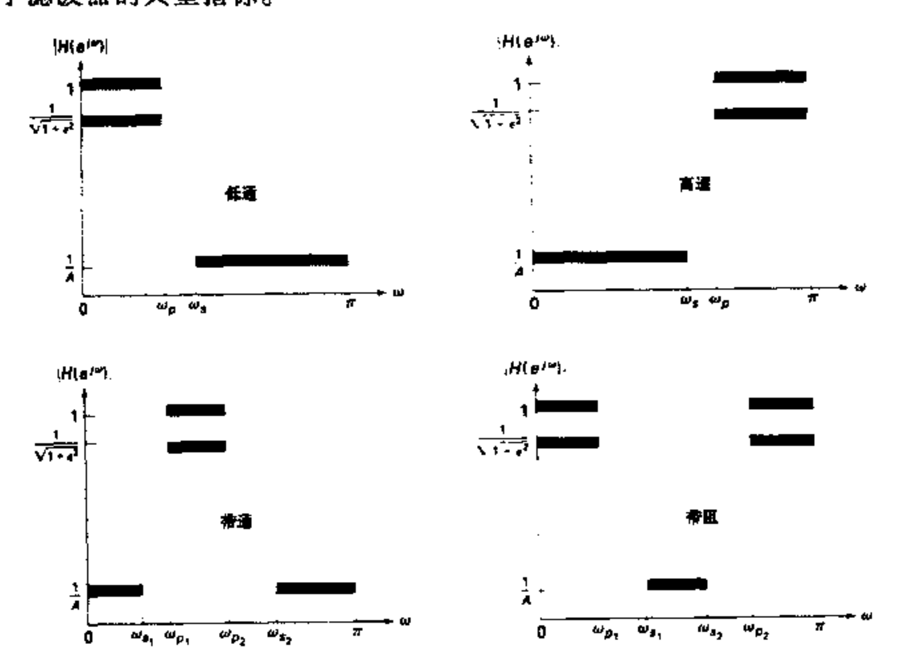


图 8.23 选频滤波器的指标

设 $H_{LP}(Z)$为给定的低通数字滤波器原型，并设 $H(z)$是期望的选频数字滤波器，注意相应于 H_{LP}和 H，选用了不同的变量 Z 和 z，定义映射形式为：

$$Z^{-1} = G(z^{-1})$$

使

$$H(z) = H_{LP}(Z)\big|_{Z^{-1}=G(z^{-1})}$$

为了做到这一点,只需把 H_{LP}中的Z^{-1}用函数 $G(z^{-1})$代换即可。假定 $H_{LP}(Z)$是稳定的和因果的滤波器,我们也希望 $H(z)$是稳定的和因果的。因此,得到下面的要求:

1.G(.)必须是 z^{-1}的有理函数,这样 $H(z)$才是可实现的。

2.Z 平面的单位圆必须映射到 z 平面的单位圆上。

3.对于稳定的滤波器,Z 平面的单位圆内部必须映射到 z 平面的单位圆内部。

设 ω' 和 ω 分别是 Z 和 z 的频率变量,即在各自单位圆 $Z=e^{j\omega'}$ 以及 $z=e^{j\omega}$ 上。则上面的要求 2 意味着

$$|Z^{-1}| = |G(z^{-1})| = |G(e^{-j\omega})| = 1$$

以及

$$e^{j\omega'} = |G(e^{-j\omega})|e^{j\angle G(e^{-j\omega})}$$

或者

$$-\omega' = \angle G(e^{-j\omega})$$

满足上述要求的函数 G(.)的一般形式是全通类型的有理函数,为:

$$Z^{-1} = G(z^{-1}) = \pm\prod_{k=1}^{n}\frac{z^{-1}-\alpha_k}{1-\alpha_k z^{-1}}$$

为了满足要求 3 的稳定性,$|\alpha_k| < 1$。

现在选择合适的阶数 n 和系数$\{\alpha_k\}$,就可以得到各种映射。表 8.2 中给出应用最广的几种映射。我们通过设计高通滤波器来说明这个表的用法。

表 8.2 数字滤波器的频带变换(原型低通滤波器的截止频率为 ω'_c)

变换类型	变换	参数
低通	$z^{-1}\to\frac{z^{-1}-\alpha}{1-\alpha z^{-1}}$	ω_c = 新滤波器的截止频率
		$\alpha=\frac{\sin[(\omega'_c-\omega_c)/2]}{\sin[(\omega'_c+\omega_c)/2]}$
高通	$z^{-1}\to\frac{z^{-1}+\alpha}{1+\alpha z^{-1}}$	ω_c = 新滤波器的截止频率
		$\alpha=-\frac{\cos[(\omega'_c+\omega_c)/2]}{\cos[(\omega'_c-\omega_c/2]}$

续表

变换类型	变换	参数
带通	$z^{-1} \to \frac{z^{-2} - \alpha_1 z^{-1} + \alpha_2}{\alpha_2 z^{-2} - \alpha_1 z^{-1} + 1}$	ω_l = 低截止频率
		ω_u = 高截止频率
		$\alpha_1 = -2\beta K/(K+1)$
		$\alpha_2 = (K-1)/(K+1)$
		$\beta_k = \frac{\cos[(\omega_u + \omega_l)/2]}{\cos(\omega_u - \omega_l)/2]}$
		$K = \cos\frac{\omega_u - \omega_c}{2}\tan\frac{\omega'_c}{2}$
带阻	$z^{-1} \to \frac{z^{-2} - \alpha_1 z^{-1} + \alpha_2}{\alpha_2 z^{-2} - \alpha_1 z^{-1} + 1}$	ω_l = 低截止频率
		ω_u = 高截止频率
		$\alpha_1 = -2\beta/(K+1)$
		$\alpha_2 = (K-1)/(K+1)$
		$\beta_k = \frac{\cos[(\omega_u + \omega_c)/2]}{\cos[(\omega_u - \omega_l)/2]}$
		$K = \tan\frac{\omega_u - \omega_l}{2}\tan\frac{\omega'_c}{2}$

例 8.25 在例8.22中,设计了一个切比雪夫Ⅰ型低通滤波器,指标为:

$$\omega'_p = 0.2\pi, \qquad R_p = 1\text{dB}$$
$$\omega'_s = 0.3\pi, \qquad A_s = 15\text{dB}$$

并且确定了它的系统函数

$$H_{LP}(Z) = \frac{0.001836(1+Z^{-1})^4}{(1-1.4996Z^{-1}+0.8482Z^{-2})(1-1.5548Z^{-1}+0.6493Z^{-2})}$$

设计一个高通滤波器,容限与上述相同,但通带从 $\omega_p = 0.6\pi$ 开始。

解:把给定的低通滤波器转换成高通滤波器,使截止频率 $\omega'_p = 0.2\pi$ 映射成 $\omega_p = 0.6\pi$ 由表 8.2 得

$$\alpha = -\frac{\cos[(0.2\pi + 0.6\pi)/2]}{\cos[(0.2\pi - 0.6\pi)/2]} = -0.38197 \tag{8.30}$$

因此

$$H_{LP}(z) = H(Z)\Big|_{Z=\frac{z^{-1}-0.38197}{1-0.38197z^{-1}}}$$

$$= \frac{0.02426(1-z^{-1})^4}{(1+0.5661z^{-1}+0.7657z^{-2})(1+1.0416z^{-1}+0.4019z^{-2})}$$

为预期的滤波器，图 8.24 给出低通滤波器和新设计的高通滤波器的频率响应图。

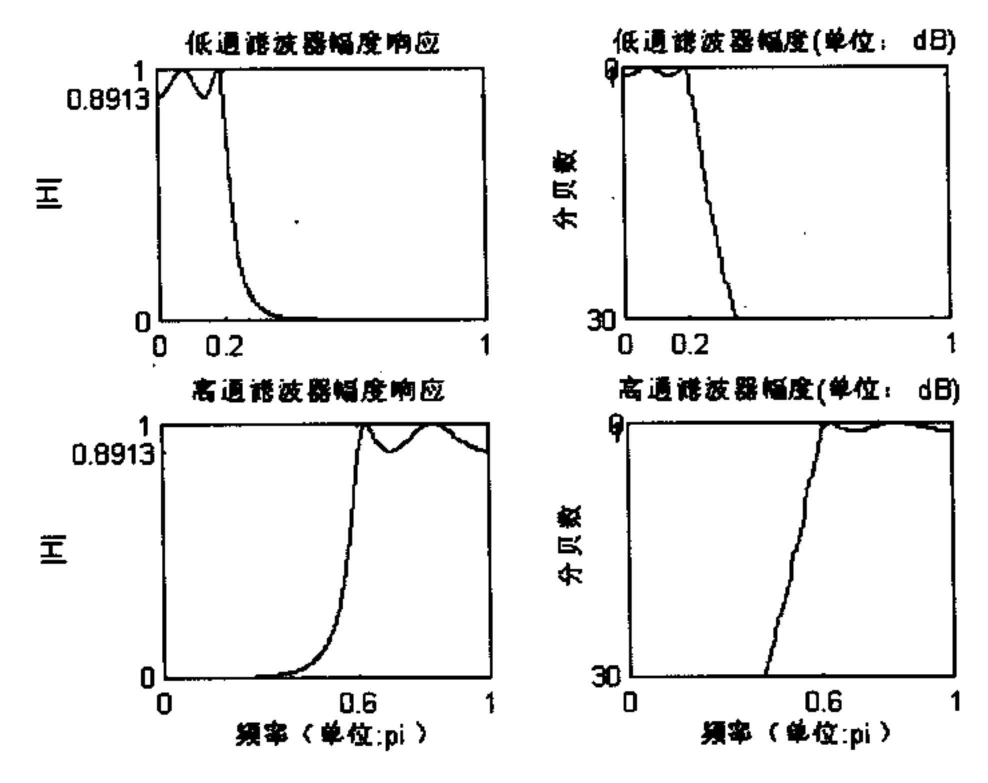


图 8.24　例 8.25 的幅度响应图

从上面的例子可以清楚地看出，为了从低通数字滤波器原型得到新数字滤波器的有理函数，应该根据表 8.2 实现有理函数代换。这似乎是一个困难的任务，但是因为这些都是代数函数，所以可以重复地使用 conv 函数。下面的 zmapping 函数说明这个方法。

```
function [bz,az] = zmapping(bZ,aZ,Nz,Dz)
% 从 Z 域到 z 域的频带变换
% -------------------------------------
% [bz,az] = zmapping(bZ,aZ,Nz,Dz)
% 实现:
%
% b(Z)        b(Z)     |
% ----- = -----        |       N(z)
% a(Z)        a(Z)     |   Z = -----
%                              D(z)
%
bzord = (length(bZ) - 1) * (length(Nz) - 1);
azord = (length(aZ) - 1) * (length(Dz) - 1);

bz = zeros(1,bzord + 1);
```

```
for k = 0:bzord
        pln = [1];
        for l = 0:k-1
                pln = conv(pln,Nz);
        end
        pld = [1];
        for l = 0:bzord-k-1
                pld = conv(pld,Dz);
        end
        bz = bz+bZ(k+1)*conv(pln,pld);
end

az = zeros(1,azord+1);
for k = 0:azord
        pln = [1];
        for l = 0:k-1
                pln = conv(pln,Nz);
        end
        pld = [1];
        for l = 0:azord-k-1
                pld = conv(pld,Dz);
        end
        az = az+aZ(k+1)*conv(pln,pld);
end
az1 = az(1); az = az/az1; bz = bz/az1;
```

例 8.26 用 zmapping 函数实现例 8.25 中的低通到高通变换。

解:首先在 MATLAB 中用双线性变换设计低通数字滤波器,然后使用 zmapping 函数。

```
>> % 数字低通滤波器指标:
>> wplp = 0.2*pi;                          % 数字通带频率(弧度)
>> wslp = 0.3*pi;                          % 数字阻带频率(弧度)
>>  Rp = 1;                                % 通带波动(dB)
>>  As = 15;                               % 阻带衰减(dB)

>> % 模拟原型指标频率的逆映射:
>> T = 1; Fs = 1/T;                        % 置 T=1
>> OmegaP = (2/T)*tan(wp/2);               % 预修正原型通带频率
>> OmegaS = (2/T)*tan(ws/2);               % 预修正原型阻带频率
```

```
>> %模拟切比雪夫原型滤波器计算:
>>[cs,ds] = afd_chb1(OmegaP,OmegaS,Rp,As);
* * * 切比雪夫Ⅰ型滤波器阶次 = 4

>> % 双线性变换:
>> [blp,alp] = bilinear(cs,ds,T);

>> % 数字高通滤波器截止频率
>> wphp = 0.6*pi;                                  % 通带边缘频率

>> % 低通-高通频带变换:
>> alpha = -(cos((wplp+wphp)/2))/(cos((wplp-wphp)/2))
alpha = -0.3820

>> Nz = -[alpha,1]; Dz = [1,alpha];
>> [bhp,ahp] = zmapping(blp,alp,Nz,Dz);
>> [C,B,A] = dir2cas(bhp,ahp)
C = 0.0243
B = 1.0000   -2.0000    1.0000
         1.0000   -2.0000    1.0000
A = 1.0000    1.0416    0.4019
         1.0000    0.5561    0.7647
```

高通滤波器的系统函数为

$$H(z) = \frac{0.0243(1 - z^{-1})^4}{(1 + 0.5661z^{-1} + 0.7647z^{-2})(1 + 1.0416z^{-1} + 0.4019z^{-2})}$$

与例 8.25 基本相同。

设计过程

在例 8.26 中低通数字滤波器原型有效地转换成了高通滤波器,正确地映射了一个特定的带边缘频率。在实际中,我们必须首先设计低通数字滤波器原型,它的技术指标应该由图8.23给出的其他类型的选频滤波器的指标得到。也可通过表 8.2 给出的变换公式得到。

以例 8.25 中的高通滤波器为例,通带边缘频率是经过(8.30)的参数 $\alpha = -0.38197$ 变换得到的。对应于低通滤波器原型的阻带边缘频率 $\omega_s' = 0.3\pi$,高通滤波器的阻带边缘频率 ω_s 是什么? 这可由(8.30)得到。由于在变换中 α 是一个常数,因此有

$$\alpha = \frac{\cos[(0.3\pi + \omega_s)/2]}{\cos[(0.3\pi - \omega_s)/2]} = -0.38197$$

这是一个超越方程，要通过猜设初值进行迭代求解。这可用 MATLAB 来实现，它的解为：

$$\omega_s = 0.4586\pi$$

在实际中，预期的高通频率 ω_s 和 ω_p 是已知的，但需要找出低通原型截止频率 ω'_s 和 ω'_p。我们可以选择一个合理的通带频率 ω'_p，比如 $\omega'_p = 0.2\pi$。利用表 8.2 中的公式从 ω_p 确定 α。现在，可以根据 α 确定 ω'_s（对于这个高通滤波器的例子来讲），并且：

$$Z = \frac{z^{-1} + \alpha}{1 + \alpha z^{-1}}$$

其中 $Z = e^{j\omega'_s}$ 以及 $z = e^{j\omega_s}$，或者

$$\omega'_s = \angle\left(-\frac{e^{-j\omega_s} + \alpha}{1 + \alpha e^{-j\omega_s}}\right) \tag{8.31}$$

继续以高通滤波器为例，设 $\omega_p = 0.6\pi$ 和 $\omega_s = 0.4586\pi$ 为频带边缘频率。假设选择 $\omega'_p = 0.2\pi$，则根据(8.30)得 $\alpha = -0.38197$，并且由(8.31)

$$\omega'_s = \angle\left(-\frac{e^{-j0.4586\pi} - 0.38197}{1 - 0.38197e^{-j-0.38197}}\right) = 0.3\pi$$

与预想的一样。现在可以设计一个数字低通滤波器，然后用 zmapping 函数把它变换成高通滤波器，以完成设计过程。为了设计一个高通切比雪夫Ⅰ型数字滤波器，上述过程可由下面的 cheb1hpf 函数具体实现。

```
function [b,a] = cheb1hpf(wp,ws,Rp,As)
% 用切比雪夫Ⅰ型为原型设计 IIR 高通滤波器
% [b,a] = cheb1hpf(wp,ws,Rp,As)
%    b = 高通滤波器的分子多项式
%    a = 高通滤波器的分母多项式
% wp = 以弧度为单位的通带频率
% ws = 以弧度为单位的阻带频率;
% Rp = 通带波动的 + dB 数; (Rp > 0)
% As = 阻带衰减的 + dB 数; (As > 0)
%
% 求出数字低通截止频率:
wplp = 0.2 * pi;
alpha = -(cos((wplp + wp)/2))/(cos((wplp - wp)/2));
```

```
wslp = angle( - (exp( - j * ws) + alpha)/(1 + alpha * exp( - j * ws)));
%
% 求出模拟低通原型指标:
T = 1; Fs = 1/T;
OmegaP = (2/T) * tan(wplp/2);
OmegaS = (2/T) * tan(wslp/2);

% 设计模拟切比雪夫原型低通滤波器:
[cs,ds] = afd_chb1(OmegaP,OmegaS,Rp,As);

% 实行双线性变换以得到数字低通:
[blp,alp] = bilinear(cs,ds,Fs);

% 将数字低通变换为高通滤波器:
Nz = -[alpha,1]; Dz = [1,alpha];
[b,a] = zmapping(blp,alp,Nz,Dz);
```

例 8.27 用切比雪夫Ⅰ型原型设计高通数字滤波器,满足

$$\omega_p = 0.6\pi, \qquad R_p = 1\text{dB}$$
$$\omega_s = 0.4586\pi, \qquad A_s = 15\text{dB}$$

解:MATLAB 程序

```
>> % 数字高通滤波器指标:
>> wp = 0.6 * pi;                       % 数字通带频率(弧度)
>> ws = 0.4586 * pi;                    % 数字阻带频率(弧度)
>> Rp = 1;                              % 通带波动(dB)
>> As = 15;                             % 阻带衰减(dB)

>> [b,a] = cheb1hpf(wp,ws,Rp,As);
>> [C,B,A] = dir2cas(b,a)
C = 0.0243
B = 1.0000   -2.0000    1.0000
    1.0000   -2.0000    1.0000
A = 1.0000    1.0416    0.4019
    1.0000    0.5561    0.7647
```

系统函数为

$$H(z) = \frac{0.0243(1 - z^{-1})^4}{(1 + 0.5661z^{-1} + 0.7647z^{-2})(1 + 1.0416z^{-1} + 0.4019z^{-2})}$$

与例 8.26 中的相同。

利用表 8.2 中的变换公式,可以很容易地把上述高通滤波器设计过程推广至其他类型的选频滤波器。这些设计过程在习题 8.18 至 8.22 中探讨。现在讨论利用 MATLAB 中的有关函数设计任意的选频滤波器。

MATLAB 实现

在前一节我们讨论了用来设计数字低通滤波器的 4 个 MATLAB 函数,这些函数也可用来设计高通、带通以及带阻滤波器。这些函数中的频带变换在 s 平面完成。以 butter 函数为例进行说明,根据输入量的不同,它有以下几种形式。

·[b,a] = BUTTER(N,wn,'high')设计 N 阶高通滤波器,wn 为它的 3 _ dB 截止频率,其单位为 π.

·[b,a] = BUTTER(N,wn,) 当 wn 为具有两个元素的矢量 wn = [w1 w2]时,它设计 2N 阶带通滤波器,它具有单位为 π 的 3 _ dB 通带 w1 < w < w2。

·[b,a] = BUTTER(N,wn,'stop') 若 wn = [w1 w2],则它设计 2N 阶带阻滤波器,它具有单位为 π 的 3 _ dB 阻带 w1 < w < w2。

为了设计任意的选频巴特沃思滤波器,必须知道阶数 N 和 3 _ dB 截止频率矢量 wn。在本章中,讨论了怎样确定低通滤波器的这些参数。但是,对于带通和带阻滤波器,计算要复杂得多。在信号处理工具箱中,MATLAB 提供了一个叫做 buttord 的函数计算这些参数。给定指标 ω_p, ω_s, R_p, 和 A_s,这个函数可求出必要的参数,它的句法为

[N,wn] = buttord(wp,ws,Rp,As)

对于不同类型的滤波器,参数 wp 和 ws 有一些限制:

- 对于低通滤波器 wp < ws;
- 对于高通滤波器 wp > ws;
- 对于带通滤波器,wp 和 ws 是具有两个元素的矢量,wp = [wp1,wp2]和 ws = [ws1,ws2],并且 ws1 < wp1 < wp2 < ws2;
- 对于带阻滤波器 wp1 < ws1 < ws2 < wp2。

把 buttord 函数和 butter 函数结合起来,就可以设计任意的巴特沃思 IIR 滤波器。经过适当修正,对 cheby1、cheby2、ellip 函数可作类似的讨论。我们将通过例子说明这些函数的用法。

例 8.28 设计切比雪夫 Ⅰ 型高通滤波器,满足例 8.27 中给定的指标。

解:MATLAB 程序————————————

```
>> % 数字低通滤波器指标:                    % 切比雪夫Ⅰ型高通:
>> ws = 0.4586*pi;                          % 数字阻带边缘频率
>> wp = 0.6*pi;                             % 数字通带边缘频率
>> Rp = 1;                                  % 通带波动(dB)
>> As = 15;                                 % 阻带衰减(dB)

>> % 计算切比雪夫Ⅰ滤波器参数
```

```
>> [N,wn] = cheb1ord(wp/pi,ws/pi,Rp,As);

>> % 数字切比雪夫Ⅰ高通滤波器设计:
>> [b,a] = cheby1(N,Rp,wn,'high');
>> % 级联形式实现:
>> [b0,B,A] = dir2cas(b,a)
b0 = 0.0243
B = 1.0000   -1.9991   0.9991
    1.0000   -2.0009   1.0009
A = 1.0000    1.0416   0.4019
    1.0000    0.5561   0.7647
```

级联形式的系统函数为

$$H(z)=\frac{0.0243(1-z^{-1})^4}{(1+0.5661z^{-1}+0.7647z^{-2})(1+1.0416z^{-1}+0.4019z^{-2})}$$

与例 8.27 中设计的滤波器相同,这证明两种方法是一样的。频域响应图如图 8.25 所示。

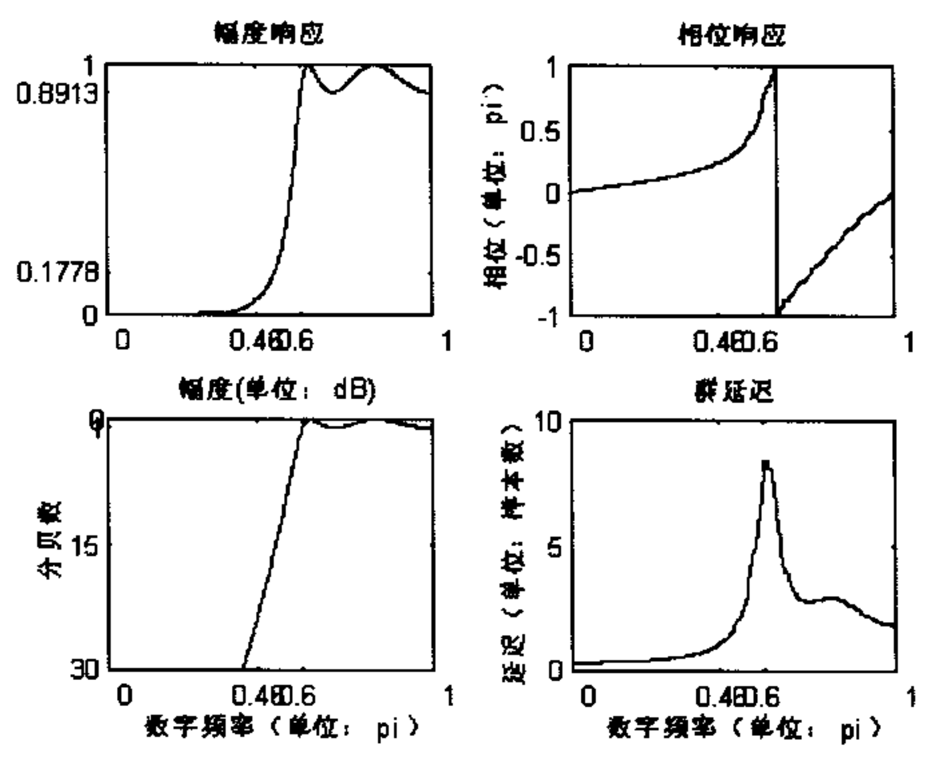


图 8.25　例 8.28 中的数字切比雪夫Ⅰ型高通滤波器

例 8.29　设计一个椭圆带通滤波器,它的指标在下面的 MATLAB 程序中给出:

```
>> % 数字低通滤波器指标:                          % 类型:椭圆型带通
>> ws = [0.3 * pi 0.75 * pi];                    % 数字阻带边缘频率
>> wp = [0.4 * pi 0.6 * pi];                     % 数字通带边缘频率
>> Rp = 1;                                       % 通带波动(dB)
>> As = 40;                                      % 阻带衰减(dB)

>> % 计算椭圆滤波器参数:
>> [N,wn] = ellipord(wp/pi,ws/pi,Rp,As);

>> % 数字椭圆滤波器设计:
>> [b,a] = ellip(N,Rp,As,wn);

>> % 级联形式实现:
>> [b0,B,A] = dir2cas(b,a)
b0 = 0.0197
B = 1.0000    1.5066    1.0000
    1.0000    0.9268    1.0000
    1.0000   -0.9268    1.0000
    1.0000   -1.5066    1.0000
A = 1.0000    0.5963    0.9399
    1.0000    0.2774    0.7929
    1.0000   -0.2774    0.7929
    1.0000   -0.5963    0.9399
```

所设计的滤波器是一个十阶滤波器,频域响应图如图 8.26 所示。

例 8.30 最后设计一个切比雪夫Ⅱ型带通滤波器,它的指标在下面的 MATLAB 中给出:

```
>> % 数字低通滤波器指标:                          % 类型:切比雪夫Ⅱ型带阻
>> ws = [0.4 * pi 0.7 * pi];                     % 数字阻带边缘频率
>> wp = [0.25 * pi 0.8 * pi];                    % 数字通带边缘频率
>> Rp = 1;                                       % 通带波动(dB)
>> As = 40;                                      % 阻带衰减(dB)

>> % 计算切比雪夫Ⅱ型滤波器参数
>> [N,wn] = cheb2ord(wp/pi,ws/pi,Rp,As);

>> % 数字切比雪夫Ⅱ型带阻滤波器设计:
>> [b,a] = cheby2(N,As,ws/pi,'stop');

>> % 级联形式实现:
```

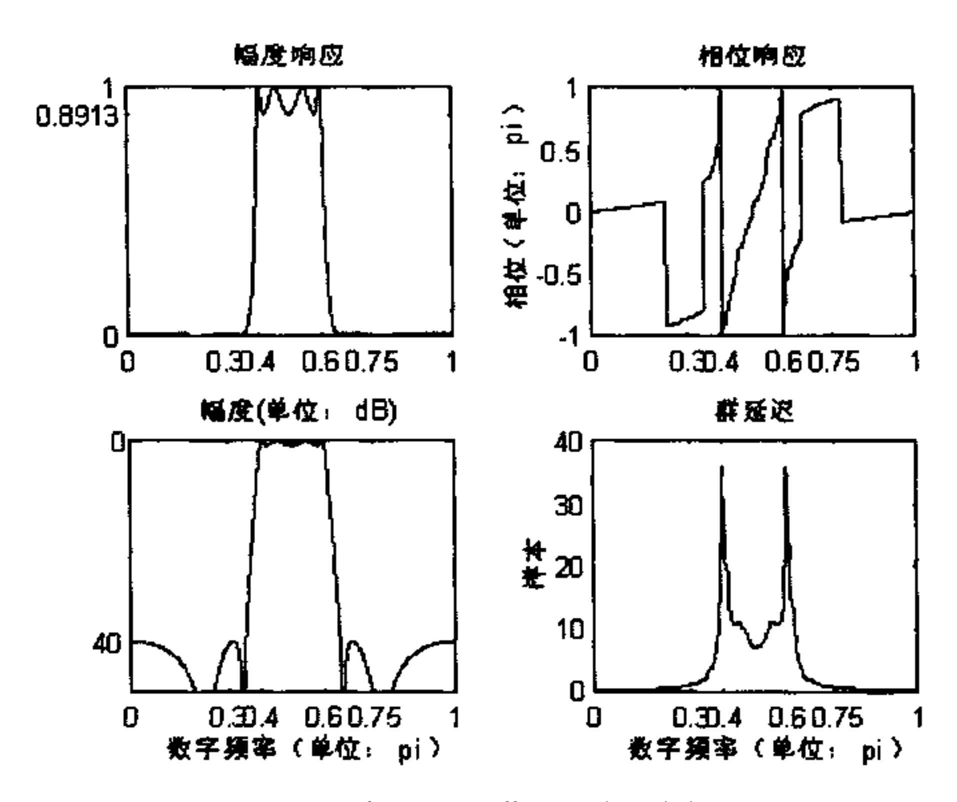


图 8.26 例 8.29 中的数字椭圆带通滤波器

```
>> [b0,B,A] = dir2cas(b,a)
b0 = 0.1558
B = 1.0000    1.1456    1.0000
    1.0000    0.8879    1.0000
    1.0000    0.3511    1.0000
    1.0000   -0.2434    1.0000
    1.0000   -0.5768    1.0000
A = 1.0000    1.3041    0.8031
    1.0000    0.8901    0.4614
    1.0000    0.2132    0.2145
    1.0000   -0.4713    0.3916
    1.0000   -0.8936    0.7602
```

这同样是一个十阶滤波器，频域响应图如图 8.27 所示。

FIR 滤波器和 IIR 滤波器的比较

至今为止，我们已经学习了许多设计 FIR 和 IIR 滤波器的技术。实际上，对于一个特定的应用，人们往往不知该选用哪种类型(FIR 或 IIR)的滤波器，该采用什么方法进行设计。由于这些设计技术中包含不同的设计机理，因此很难对它们进行比较。但是若将注意力集中在最大最小化最优(或等波动)滤波器，就可以进行一些有意义的比较。对于 FIR 滤波器，这些

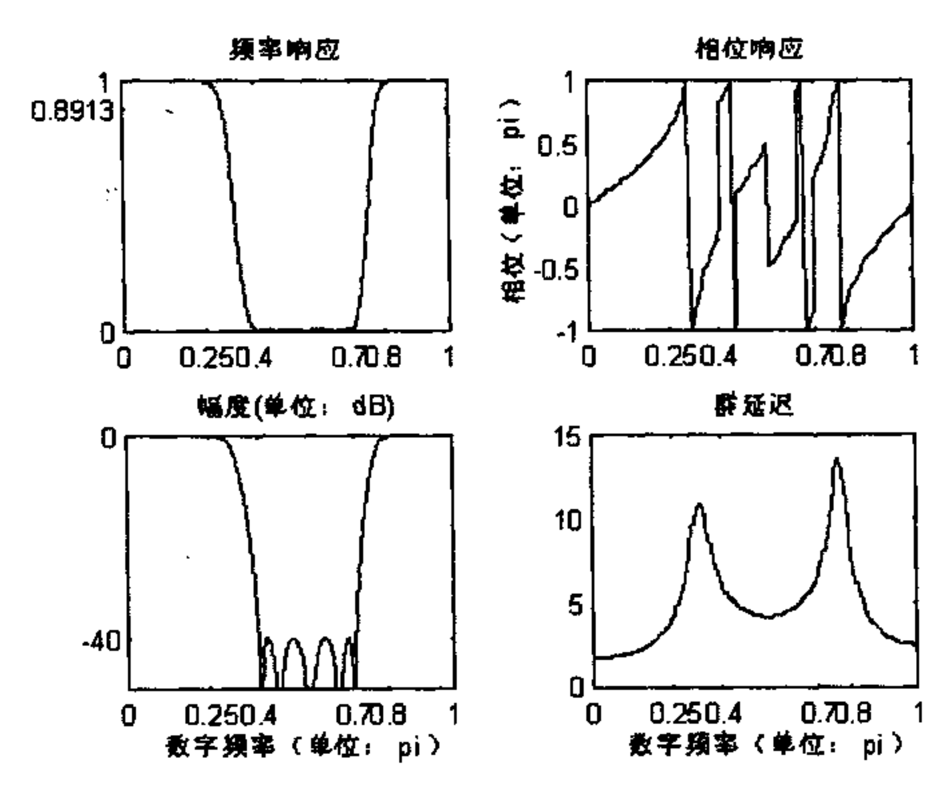


图 8.27 例 8.30 中的切比雪夫Ⅱ型带通滤波器

最优滤波器是由 Parks-McClellan 算法(或 Remez 交换算法)设计的等波动滤波器,而对于 IIR 滤波器来讲,是椭圆滤波器。

比较的一个基础是这些滤波器在标准实现下,计算一个输出样本所需要的乘法次数。对于 FIR 滤波器,标准实现是线性相位直接形式,而对于椭圆滤波器,经常使用级联形式。设 M 为线性相位 FIR 滤波器的长度(假定 M 为奇数),则每个输出样本需要

$$\frac{M+1}{2} \simeq \frac{M}{2} \qquad M \text{ 较大时}$$

次乘法。设 N(假定 N 为偶数)是以级联形式实现的椭圆滤波器的阶数,因此它有 $N/2$ 个二阶子系统,每个子系统需要 3 次乘法(在最经济实现的情况下),整个结构需要 3 次附加的乘法,因此每个输出样本需要

$$3\frac{N}{2}+3 \simeq \frac{3N}{2} \qquad N \text{ 较大时}$$

次乘法。

现在假设每个滤波器都能精确地符合相同的指标(例如,低通滤波器的 ω_p, ω_s, δ_1(或通带波动 R_p),以及 δ_2(或阻带衰减 A_s)),则若

$$\frac{M+1}{2} = \frac{3N+3}{2} \Rightarrow \frac{M}{N} = 3+\frac{1}{N} \simeq 3 \qquad N \text{ 较大时}$$

这两个滤波器等价。这说明如果比率 $M/N=3$,则这两个滤波器近似同等有效。但是若 $M/N<3$,等波动 FIR 滤波器更有效,或者若$M/N>3$时,椭圆 IIR 滤波器更有效。

实验表明:

- $\omega_p \geqslant 0.3$ 时,对于所有的 δ_1,δ_2,N, $M/N\geqslant 3$
- $N\geqslant 10$ 时,对于所有的 $\delta_1,\delta_2,N,M/N\geqslant 3$
- N 较大时,M/N 的数量级为百

这说明对于大多数应用场合,从计算的观点来看,IIR 椭圆滤波器是较好的选择。最适于 FIR 滤波器的情况是:

- 大的 δ_1 值
- 小的 δ_2 值
- 大的过渡带宽

此外,由于线性相位特性的要求,往往须在椭圆滤波器中级联相位均衡器(它是全通滤波器),由于 FIR 等波动滤波器具有的精确线性相位特性,用它看来是不错的。

习　　题

P8.1 设计一个模拟巴特沃思低通滤波器,它在 30 弧度/秒处具有 1dB 或更好的波动,在 40 弧度/秒处具有至少 30dB 的衰减。求出级联形式的系统函数,画出滤波器的幅度响应、对数幅度响应、相位响应和脉冲响应图。

P8.2 设计低通模拟椭圆滤波器,它具有下列特征:

·可接受的 1dB 通带波动;

·通带截止频率为 10 弧度/秒;

·在 15 弧度/秒以外,阻带衰减为 40dB 或更大。

求出有理函数形式的系统函数,画出滤波器的幅度响应、对数幅度响应、相位响应和脉冲响应图。

P8.3 信号 $x_a(t)$包含两个频率,100Hz 和 130Hz。我们希望抑制 130Hz 的分量至 50dB 衰减,同时以小于 2dB 的衰减通过 100Hz 的分量。设计一个具有最小阶数的切比雪夫Ⅰ型模拟滤波器实现此滤波操作,画出对数幅度响应图并验证你的设计。

P8.4 设计一个模拟切比雪夫Ⅱ型低通滤波器,它在 250Hz 处具有 0.5dB 或更好的波动,在 300Hz 处至少有 45dB 的衰减。画出滤波器的幅度响应、对数幅度响应、相位响应和脉冲响应图。

P8.5 写出一个用来设计模拟低通滤波器的 MATLAB 函数,函数的格式应为:

```
function [b,a] = afd(type,Fp,Fs,Rp,As)
%
% function [b,a] = afd(type,Fp,Fs,Rp,As)
% 模拟低通滤波器设计
% type='butter'或'cheby1'或'cheby2'或'ellip'
% Fp = 以 Hz 为单位的通带截止频率
% Fs = 以 Hz 为单位的阻带截止频率
```

% Rp = 通带波动的 dB 数

% As = 阻带衰减的 dB 数

利用本章中开发的 afd _ butt,afd _ chb1,afd _ chb2 和 afd _ elip 函数实现。并用习题 8.1 至 8.4 中给定的指标验证你的函数。

P8.6 设计一个用在结构

$$x_a(t)\rightarrow\boxed{A/D}\rightarrow\boxed{H(z)}\rightarrow\boxed{D/A}\rightarrow y_a(t)$$

中的低通数字滤波器,满足下列要求:

- 采样速率为 8000 个样本/秒;
- 通带边缘频率为 1500Hz,波动为 3dB;
- 阻带边缘频率为 2000Hz,衰减为 40dB;
- 通带等波动但是阻带是单调的;
- 采用脉冲响应不变法;

a. 在脉冲响应不变方法中,选择 $T=1$,求出并联形式的系统函数 $H(x)$,画出对数幅度响应和脉冲响应 $h(n)$图。

b. 在脉冲响应不变法中,选择 $T=1/8000$,求出并联形式的系统函数 $h_a(s)$,画出对数幅度响应和脉冲响应 $h(n)$图,比较上述两种设计,讨论 T 对设计的影响。

P8.7 设计巴特沃思数字低通滤波器满足下列指标:

通带边缘频率:0.4π, $R_p = 0.5\text{dB}$

阻带边缘频率:0.6π, $A_s = 50\ \text{dB}$

采用脉冲响应不变方法,$T=2$。求出有理函数形式的系统函数,画出对数幅度响应、脉冲响应 $h(n)$和模拟原型滤波器的脉冲响应 $h_a(t)$图,并比较它们的形状。

P8.8 写出基于脉冲响应不变变换,设计数字低通滤波器的 MATLAB 函数,函数的格式应为:

```
function [b,a] = dlpfd_ii(type,wp,ws,Rp,As,T)
%
% function [b,a] = dlpfd_ii(type,wp,ws,Rp,As,T)
% 用脉冲响应不变技术设计数字低通滤波器
% type='butter'或'cheby1'
% wp = 以弧度为单位的通带截止频率
% ws = 以弧度为单位的阻带截止频率
% Rp = 通带波动的 dB 数
% As = 阻带衰减的 dB 数
% T=采样间隔
```

利用习题 8.5 中开发的 afd 函数。并用习题 8.6 至 8.7 中的指标验证此函数。

P8.9 在本习题中,我们将研究阶跃不变变换技术。经过这种变换,所设计的数字滤波器保持模拟滤波器原型的阶跃响应不变;也就是说,若 $\xi_a(t)$是模拟原型的阶跃响应,$\xi(n)$是数字滤波器的阶跃响应,则

$$\xi(n) = \xi_a(t = nT),\ T\text{:采样间隔}$$

在频域中的关系为：

$$\Xi_a(s) \triangleq L[\xi_a(t)] = H_a(s)/s$$

并且

$$\Xi(z) \triangleq Z[\xi(n)] = H(z)\frac{1}{1-z^{-1}}$$

则阶跃'不变变换的步骤如下：给定 $H_a(s)$

·将 $H_a(s)$除以 s 得到$\Xi_a(s)$；

·找出 $\Xi_a(s)$的极点$\{pk\}$和留数$\{R_k\}$；

·把模拟极点$\{pk\}$变换成数字极点$\{e^{pkT}\}$，其中 T 为任意；

·根据留数$\{R_k\}$和极点$\{e^{pkT}\}$求出 $\Xi(z)$；

·将 $\Xi(z)$乘以$(1-z^{-1})$得到 $H(z)$；

根据上述过程开发一个实现阶跃不变变换的 MATLAB 函数. 函数的格式应为

```
function[b,a] = stp_invr(c,d,T)
% 从模拟滤波器到数字滤波器的阶跃不变变换
% function[b,a] = stp_invr(c,d,T)
% b = 自变量为 z^(-1)的数字滤波器的分子多项式
% a = 自变量为 z^(-1)的数字滤波器的的分母多项式
% c = 自变量为 s 的模拟滤波器的分子多项式
% d = 自变量为 s 的模拟滤波器的分母多项式
% T = 采样(变换)参数
```

P8.10 用阶跃不变法设计习题 8.7 中的低通巴特沃思数字滤波器，画出对数幅度响应并与习题 8.7 比较，画出阶跃响应 $\xi(n)$和模拟原型的脉冲响应 $\xi_a(t)$，并比较它们的形状。

P8.11 考虑习题 8.7 中的低通巴特沃思滤波器设计

a. 利用本章讨论的双线性变换技术原理以及 bilinear 函数，画出对数幅度响应比较模拟原型和数字滤波器的脉冲响应。

b. 利用 butter 函数进行设计，并与 a 中的设计进行比较。

P8.12 利用双线性变换法设计习题 8.6 中的模拟切比雪夫Ⅰ型滤波器，并对两种设计进行比较。

P8.13 用椭圆原型设计数字低通滤波器，满足下列要求：

通带边缘频率：0.4π，　　$R_p = 1$ dB

阻带边缘频率：0.5π，　　$A_s = 60$ dB

用 ellip 函数和 bilinear 函数进行设计，并对结果进行比较。

P8.14 设计数字低通滤波器，以满足下列要求：

通带边缘频率：0.3π，　　$R_p = 0.5$dB

阻带边缘频率：0.4π，　　$As = 50$ dB

a.利用 butter 函数进行设计,求出阶数 N 和实际的最小阻带衰减的 dB 数。

b.利用 cheby1 函数进行设计,求出阶数 N 和实际的最小阻带衰减的 dB 数。

c.利用 cheby2 函数进行设计,求出阶数 N 和实际的最小阻带衰减的 dB 数。

d.利用 ellip 函数进行设计,求出阶数 N 和实际的最小阻带衰减的 dB 数。

e.比较上述各种设计的阶数、实际最小阻带衰减以及群延迟。

P8.15 利用本章讨论的原理,写出一个 MATLAB 函数,它根据高通数字滤波器的指标求出低通原型数字滤波器的各指标,函数的格式应为:

```
function [wpLP,wsLP,alpha] = hp2lpfre(wphp,wshp)
%从高通到低通数字滤波器的带边缘频率转换
%[wpLP,wsLP,alpha] = hp2lpfre(wphp,wshp)
% wpLP = 低通原型的通带边缘频率
% wsLP = 低通原型的阻带边缘频率
% alpha = 低通到高通的转换参数
% wphp = 高通滤波器的通带边缘频率
% wshp = 高通滤波器的阻带边缘频率
```

以此函数为基础,开发一个 MATLAB 函数,它用双线性变换设计高通数字滤波器,函数的格式应为:

```
function [b,a] = dhpfd_bl(type,wp,ws,Rp,As)
%用双线性变换设计 IIR 高通滤波器
%[b,a] = dhpfd_bl(type,wp,ws,Rp,As)
% type = 'butter'或'cheby1'或'cheby2'或'ellip'
% b = 高通滤波器的分子多项式
% a = 高通滤波器的分母多项式
% wp = 以弧度为单位的通带频率
% ws = 以弧度为单位的阻带频率(wp < ws)
% Rp = 通带中的振幅波动的 + dB 数
% As = 阻带衰减的 + dB 数
```

用例 8.27 中的指标验证你的函数。

P8.16 设计一个高通滤波器,以满足指标:

通带边缘频率:0.4π, $R_p = 0.5$dB

阻带边缘频率:0.6π, $A_s = 60$ dB

a.用习题 8.16 中的 dhpfd_bl 函数和椭圆原型设计此滤波器,画出所设计滤波器的对数幅度响应图。

b.用 ellip 函数进行设计,画出对数幅度响应图,并对上述两种设计进行比较。

P8.17 参照用于高通滤波器的方法,写出一个 MATLAB 函数,它利用表 8.2 的结果,可以从任意低通数字滤波器的指标中求出低通原型数字滤波器的指标。函数的格式应为:

```
function [wpLP,wsLP,alpha] = lp2lpfre(wplp,wslp)
%从低通到低通数字滤波器的带边缘频率转换
%[wpLP,wsLP,alpha] = lp2lpfre(wplp,wslp)
```

% wpLP = 低通原型的通带边缘频率
% wsLP = 低通原型的阻带边缘频率
% alpha = 低通到低通的转换参数
% wplp = 低通滤波器的通带边缘频率
% wslp = 低通滤波器的阻带边缘频率
以此函数为基础,开发一个 MATLAB 函数,它用双线性变换从低通数字滤波器原型设计带通滤波器,函数的格式应为:

```
function [b,a] = dbpfd_bl(type,wp,ws,Rp,As)
%用双线性变换设计 IIR 带通滤波器
%[b,a] = dbpfd_bl(type,wp,ws,Rp,As)
% type = 'butter'或'cheby1'或'cheby2'或'ellip'
% b =带通滤波器的分子多项式
% a =带通滤波器的分母多项式
% wp = 以弧度为单位的通带频率矢量[wp_lower,wp_upper]
% ws = 以弧度为单位的阻带频率矢量[ws_lower,ws_upper]
% Rp = 通带波动的 dB 数
% As = 阻带衰减的 dB 数
```

用习题 8.15 中的设计验证此函数。

P8.18 用 cheby2 函数设计一个带通数字滤波器,指标为:

低阻带边缘频率:0.3π
高阻带边缘频率:0.6π　　$A_s = 50$ dB
低通带边缘频率:0.4π　　$R_p = 0.5$dB
高通带边缘频率:0.5π

画出所设计滤波器的脉冲响应和对数幅度响应(单位:dB)。

P8.19 参照用于高通滤波器的方法,写出一个 MATLAB 函数,它利用表 8.2 的结果,可以从带通数字滤波器的指标中求出低通原型数字滤波器的指标。函数的格式应为:

```
function [wpLP,wsLP,alpha] = bp2lpfre(wpbp,wsbp)
%从带通到低通数字滤波器的带边缘频率转换
%[wpLP,wsLP,alpha] = bp2lpfre(wpbp,wsbp)
% wpLP = 低通原型的通带边缘频率
% wsLP = 低通原型的阻带边缘频率
% alpha = 低通到带通的转换参数
% wpbp = 带通滤波器的通带频率矢量[wp_lower,wp_upper](弧度)
% wsbp = 带通滤波器的阻带频率矢量[ws_lower,ws_upper](弧度)
```

以此函数为基础,开发一个 MATLAB 函数,它用双线性变换从原型低通数字滤波器设计带通滤波器,函数的格式应为:

```
function [b,a] = dbpfd_bl(type,wp,ws,Rp,As)
%用双线性变换设计 IIR 带通滤波器
%[b,a] = dbpfd_bl(type,wp,ws,Rp,As)
```

% type = 'butter'或'cheby1'或'cheby2'或'ellip'
% b = 带通滤波器的分子多项式
% a = 带通滤波器的分母多项式
% wp = 以弧度为单位的通带频率矢量[wp_lower, wp_upper]
% ws = 以弧度为单位的阻带频率矢量[ws_lower, ws_upper]
% Rp = 通带波动的 dB 数
% As = 阻带衰减的 dB 数
用习题 8.19 中的设计验证你的函数。

P8.20 用切比雪夫Ⅰ型设计一个带通 IIR 数字滤波器,满足下列指标:

$$0.95 \leqslant |H(e^{j\omega})| \leqslant 1.05, \qquad 0 \leqslant |\omega| \leqslant 0.25\pi$$
$$0 \leqslant |H(e^{j\omega})| \leqslant 0.01, \qquad 0.35\pi \leqslant |\omega| \leqslant 0.65\pi$$
$$0.95 \leqslant |H(e^{j\omega})| \leqslant 1.05, \qquad 0.75\pi \leqslant |\omega| \leqslant \pi$$

利用 cheby1 函数进行设计,求出滤波器的系统函数 $H(z)$,画出对数幅度响应图和脉冲响应图。

P8.21 参照用于高通滤波器的方法,写出一个 MATLAB 函数,它利用表 8.2 结果,可以从带阻数字滤波器的指标中求出低通原型数字滤波器的指标。函数的格式应为:

```
function [wpLP, wsLP, alpha] = bs2lpfre(wpbp, wsbp)
%从带阻到低通数字滤波器的带边缘频率转换
%[wpLP, wsLP, alpha] = bs2lpfre(wpbp, wsbp)
% wpLP = 低通原型的通带边缘频率
% wsLP = 低通原型的阻带边缘频率
% alpha = 低通到带阻的转换参数
% wpbp = 带阻滤波器的通带边缘频率矢量[wp_lower, wp_upper](弧度)
% wsbp = 带阻滤波器的阻带边缘频率矢量[ws_lower, ws_upper](弧度)
```

以此函数为基础,开发一个 MATLAB 函数,它用双线性变换从原型低通数字滤波器设计带阻滤波器,函数的格式应为:

```
function [b,a] = dbsfd_bl(type, wp, ws, Rp, As)
%用双线性变换设计 IIR 带阻滤波器
%[b,a] = dbsfd_bl(type, wp, ws, Rp, As)
% type = 'butter'或'cheby1'或'cheby2'或'ellip'
% b = 带阻滤波器的分子多项式
% a = 带阻滤波器的分母多项式
% wp = 以弧度为单位的通带频率矢量[wp_lower, wp_upper]
% ws = 以弧度为单位的阻带频率矢量[ws_lower, ws_upper]
% Rp = 通带波动的 + dB 数
% As = 阻带衰减的 + dB 数
```

用习题 8.21 中的设计验证此函数。

P8.22 模拟信号

$$x_a(t) = 5\sin(200\pi t) + 2\cos(300\pi t)$$

由系统

$$x_a(t) \rightarrow \boxed{A/D} \rightarrow \boxed{H(z)} \rightarrow \boxed{D/A} \rightarrow y_a(t)$$

处理,采样间隔为 1000 个样本/秒。

a.设计一个最小阶数的 IIR 数字滤波器,以小于 1dB 的衰减通过 150Hz 的分量,以至少 40dB 抑制 100Hz 的分量。滤波器应有单调的通带和等波动的阻带,求出有理函数形式的系统函数,画出对数幅度响应。

b.产生上述信号 $x_a(t)$的 300 个样本(采样速率为 1000 个样本/秒),通过上述所设计的滤波器得到输出序列,内插此序列(用第三章讨论的任何的内插技术)得到 $y_a(t)$。画出输入和输出信号的图并解释你的结果。

P8.23 利用双线性变换方法,设计一个带宽为 0.08π 的十阶椭圆阻带滤波器以滤除数字频率为 $\omega = 0.44\pi$ 的信号,选择合理的阻带衰减值,画出幅度响应。使序列

$$x(n) = \sin[0.44\pi n],\ n = 0, \ldots, 200$$

的 201 个样本,通过此带阻滤波器,解释所得的结果。

P8.24 设计一个数字高通滤波器 $H(z)$,它用在结构

$$x_a(t) \rightarrow \boxed{A/D} \rightarrow \boxed{H(z)} \rightarrow \boxed{D/A} \rightarrow y_a(t)$$

中,满足下列要求:

·采样速率为 10kHz;

·阻带边缘频率为 1.5kHz,衰减为 40dB;

·通带边缘频率为 2kHz,波动为 3dB;

·单调的通带和阻带;

·采用脉冲响应不变变换;

a.画出整个模拟滤波器在[0,5kHz]上的幅度响应;

b.画出数字低通原型的幅度响应;

c.画出模拟低通原型的幅度响应;

d.输入信号应有什么限制,使得上述结构能对它真正起高通滤波器的作用?

第九章　自适应滤波方面的应用

在第七、八两章中，我们叙述了 FIR 和 IIR 数字滤波器的设计方法。其目的是决定数字滤波器的系数，以使之满足某些预期的技术指标。

与前两章讨论的滤波器设计技术相反，许多数字信号处理应用中的滤波器系数不能预先确定。例如，考虑由电话信道传输数据的高速调制解调器(modem)的设计问题。这种调制解调器采用一个称为信道均衡器的滤波器，用它补偿信道失真，调制解调器必须通过具有不同的频响特性因而产生不同失真的信道有效地传输数据。使这种要求成为可能的唯一方法是信道均衡器具有可调的系数，我们能够根据信道特性测算量对这些系数进行优化，以使信道失真的某些量度最小化。这种具有可调参数的滤波器称为自适应滤波器，在调制解调器中称为自适应均衡器。

自适应滤波器的各种应用在文献中已有介绍，更值得注意的一些应用包括：(1)自适应天线系统，其中自适应滤波器用于波束方向控制，并可在波束方向图中提供一个零点以便消除不希望的干扰[23]；(2)数字通信接收机，其中自适应滤波器用于信道识别并提供码间串扰的均衡器[18]；(3)自适应噪声对消技术，其中自适应滤波器用于估计并对消预期信号中的噪声分量；(4)系统建模，其中自适应滤波器作为估计未知系统特性的模型，上述这些只是有关自适应滤波器应用方面几个最常见的例子。

虽然对自适应滤波，IIR 和 FIR 两种形式都可考虑，但 FIR 滤波器至今是实际应用最广泛的。偏爱的理由很简单，FIR 滤波器仅有可调零点，因此它没有 IIR 滤波器因兼有可调的极点和零点而带来的稳定性问题。然而，我们不能断定自适应 FIR 滤波器总是稳定的，正好相反，滤波器稳定性的关键取决于调整其系数的算法。

在我们可能用到的各种 FIR 滤波器结构中，直接型和格型是自适应滤波应用中最常用的结构形式。图 9.1 给出具有可调系数 $h(0)$，$h(1)$，…，$h(N-1)$的直接型 FIR 结构。另外，在 FIR 格型结构中的可调系数是图 6.18 所示的反射系数 K_n。

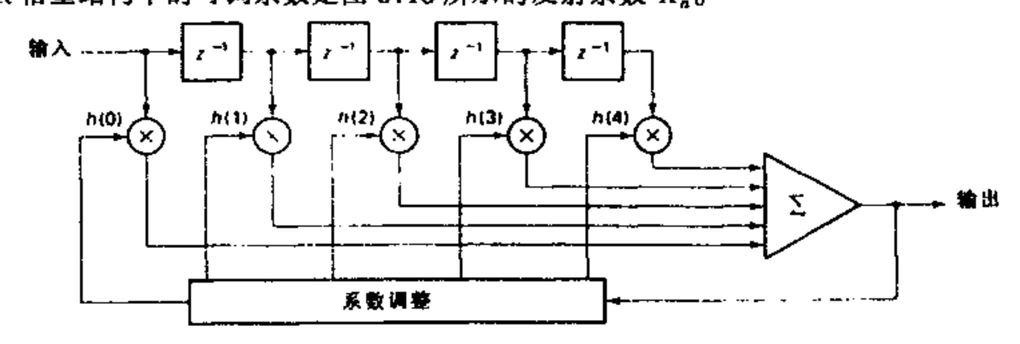


图 9.1　直接型自适应 FIR 滤波器

自适应滤波器使用中要考虑的重要问题是优化调整滤波器参数的准则，该准则不仅要对滤波器性能提供有意义的度量，而且必须导出可实现的算法。

一种可对自适应滤波应用性能提供良好度量的准则是最小平方准则及其统计描述的对等物,均方误差(MSE)准则。最小平方(和 MSE)准则导出的性能指标是滤波器系数的二次函数,它只有单个极小值。因此导出的调整滤波器系数的算法比较容易实现。

本章介绍一种自适应调整 FIR 滤波器系数的基本算法(称为最小均方(LMS)算法)。自适应滤波器的结构是具有可调系数 $h(0)$, $h(1)$,…,$h(N-1)$的直接型 FIR 滤波器,如图 9.1 所示。在介绍了 LMS 算法后,我们将其应用于几种具有自适应滤波器的实际系统。

系数调整 LMS 算法

假定我们有一个 FIR 滤波器,其可调整系数为$\{h(k),\ 0\leqslant k\leqslant N-1\}$ 设$\{X(n)\}$表示滤波器的输入序列,相应的输出序列为$\{y(n)\}$,其中

$$y(n)=\sum_{k=0}^{N-1}h(k)x(n-k),\quad n=0,\ldots,M \tag{9.1}$$

假设我们也有一个预期输出序列$\{d(n)\}$,将 FIR 滤波器输出$\{y(n)\}$与$\{d(n)\}$进行比较。于是就可以取 $d(n)$与 $y(n)$之差形成误差序列$\{e(n)\}$。即,

$$e(n)=d(n)-y(n),n=0,\ldots,M \tag{9.2}$$

选择 FIR 滤波器系数使平方误差和最小,这样,我们有

$$\begin{aligned}\varepsilon &= \sum_{n=0}^{M}e^2(n)=\sum_{n=0}^{M}\Big[d(n)-\sum_{k=0}^{N-1}h(k)x(n-k)\Big]^2 \\ &= \sum_{n=0}^{M}d^2(n)-2\sum_{k=0}^{N-1}h(k)r_{dx}(k)+\sum_{k=0}^{N-1}\sum_{l=0}^{N-1}h(k)h(l)r_{xx}(k-l)\end{aligned} \tag{9.3}$$

其中,根据定义:

$$r_{dx}(k)=\sum_{n=0}^{M}d(n)x(n-k),0\leqslant k\leqslant N-1 \tag{9.4}$$

$$r_{xx}(k)=\sum_{n=0}^{M}x(n)x(n+k),0\leqslant k\leqslant N-1 \tag{9.5}$$

我们称$\{r_{dx}(k)\}$为预期输出序列$\{d(n)\}$与输入序列$\{x(n)\}$的互相关,称$\{r_{xx}(k)\}$ 为序列$\{x(n)\}$的自相关。平方误差和 ε 是 FIR 滤波器系数的二次函数,因此它关于滤波器系数$\{h(k)\}$的最小化将导致一组线性方程,通过取 ε 对每个系数的导数,可得到

$$\frac{\partial\varepsilon}{\partial h(m)}=0,\quad 0\leqslant m\leqslant N-1 \tag{9.6}$$

所以

$$\sum_{k=0}^{N-1} h(k) r_{xx}(k-m) = r_{dx}(m), \quad 0 \leqslant m \leqslant N-1 \tag{9.7}$$

这是给出最佳滤波器系数的一组线性方程。

为了直接求解上面的线性方程组，必须先计算输入信号的自相关序列$\{r_{xx}(k)\}$和预期输出序列$\{d(n)\}$与输入序列$\{x(n)\}$的互相关序列$\{r_{dx}(k)\}$。

LMS算法提供了另一种计算方法来确定最佳滤波器系数$\{h(k)\}$，而不用明确地计算相关序列$\{r_{xx}(k)\}$和$\{r_{dx}(k)\}$。该算法基本上是一个递归梯度（最陡下降）方法，它可求出ε的最小值，并得到相应的最佳滤波器系数组。

我们从任意选择的一组$\{h(k)\}$初始值开始（如$\{h_0(k)\}$。例如，可以由$h_0(k)=0,\ 0 \leqslant k \leqslant N-1$开始，接着在每个新的输入采样值$\{x(n)\}$进入自适应滤波器后，计算相应的输出（$\{y(n)\}$），再形成误差信号$e(n)=d(n)-y(n)$，并根据如下方程不断修正滤波器系数：

$$h_n(k) = h_{n-1}(k) + \triangle \cdot e(n) \cdot x(n-k), \quad 0 \leqslant k \leqslant N-1, n = 0,1,\ldots \tag{9.8}$$

其中△为步长参数，$x(n-k)$为n时刻输入信号在滤波器的第k个抽头处的采样值，$e(n)x(n-k)$是滤波器第k个系数的负梯度的近似（估计）值。这就是自适应地调整滤波器系数以便使平方误差和ε最小化的LMS递归算法。

步长参数△控制该算法达到最佳解的收敛速度。Δ大使收敛较快，然而，如果Δ取得太大，算法会变得不稳定，为了保证稳定，Δ的取值必须按下式选择[18]

$$0 < \Delta < \frac{1}{10NP_x} \tag{9.9}$$

其中N为自适应FIR滤波器长度，P_x为输入信号功率，P_x可近似为：

$$P_x \approx \frac{1}{1+M} \sum_{n=0}^{M} x^2(n) = \frac{r_{xx}(0)}{M+1} \tag{9.10}$$

等式(9.9)和(9.10)的正确性以及导出最佳滤波器系数的LMS算法的数学证明在较高深的自适应滤波器论著中给出。有兴趣的读者可参考Haykin编写的书[9]和Proakis的书[18]。

MATLAB实现

LMS算法(9.8)可容易地用MATLAB程序实现。如果给定输入序列$\{x(n)\}$，预期序列$\{d(n)\}$，步长△和要求的自适应FIR滤波器长度N，我们就能够用(9.1)、(9.2)和(9.8)的迭代来确定自适应滤波器系数$\{h(n), 0 \leqslant n \leqslant N-1\}$，下面给出实现这种算法的MATLAB函

数,函数名为 lms。

```
function [h,y] = lms(x,d,delta,N)
%系数调整的 LMS 算法
% ----------------------------
% [h,y] =lms(x,d,delta,N)
%      h =估计的 FIR 滤波器
%      y = 输出数组 y(n)
%      x =输入数组 x(n)
%      d = 预期数组 d(n), 其长度应与 x 相同
% delta = 步长
%      N = FIR 滤波器的长度
%
M = length(x); y = zeros(1,M);
h = zeros(1,N);
for n = N:M
      x1 = x(n:-1:n-N+1);
      y = h * x1';
      e = d(n) - y;
      h = h + delta*e*x1;
end
```

另外,lms 函数还给出自适应滤波器的输出$\{y(n)\}$。

下面我们将 LMS 算法用于包含自适应滤波器的几种实际应用中。

系统辨识或系统建模

为了提出问题,让我们参阅图 9.2。有一个待识别的未知线性系统,该未知系统可以是全零点(FIR)系统,或是极点-零点(IIR)系统。要用一个长度为 N 的 FIR 滤波器来近似(模拟)这一未知系统。未知系统和 FIR 模型并联连接且用相同的输入序列$\{x(n)\}$激励。如果$\{y(n)\}$表示 FIR 模型的输出,$\{d(n)\}$表示未知系统的输出,则误差序列为$\{e(n)=d(n)-y(n)\}$。如果我们使平方误差和最小化,就得到与(9.7)相同的线性方程组。所以,(9.8)给出的 LMS 算法可以被用来不断调整 FIR 模型系数,以便其输出逼近未知系统的输出。

设计题 9.1 系统辨识

实现本题目所需要的程序模块有三个:

1.噪声信号产生器 产生零均值随机数序列。例如,产生在区间$[-a,a]$上均匀分布的随机数序列,这一均匀分布的随机数序列具有零均值和方差 $a^2/3$。该序列(称为$\{x(n)\}$)将作为未知系统和自适应 FIR 模型的输入。这时,输入信号$\{x(n)\}$的功率为 $P_x=a^2/3$。在

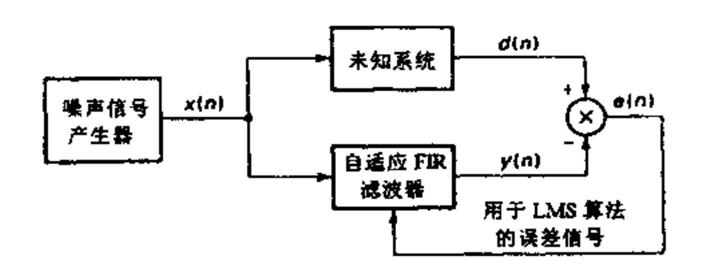


图 9.2　系统辨识或系统建模问题框图

MATLAB 中可用随机函数实现这一要求。

2.未知系统部分　选择 IIR 滤波器，用其差分方程实现。例如，可以选择由如下二阶差分方程确定的 IIR 滤波器

$$d(n)=a_1d(n-1)+a_2d(n-2)+x(n)+b_1x(n-1)+b_2x(n-2) \tag{9.11}$$

其中参数 $\{a_1,a_2\}$ 决定滤波器极点位置，$\{b_1,b_2\}$ 决定滤波器零点位置。这些参数是程序的输入变量。可用 filter 函数实现这部分。

3.自适应 FIR 滤波器程序块　FIR 滤波器有 N 个用 LMS 算法调整的抽头系数，滤波器长度 N 作为程序输入变量。可用前一节给出的 lms 函数实现自适应 FIR 滤波功能。

上述三个功能块组成如图 9.2 所示的系统，由此题我们可测定 LMS 算法收敛后 FIR 模型的单位脉冲响应逼近未知系统单位脉冲响应的程度。

为了监测 LMS 算法的收敛速度，我们可以计算平方误差 $e(n)$ 的短时平均并绘制其曲线图，即计算：

$$\mathrm{ASE}(m)=\frac{1}{K}\sum_{k=n+1}^{n+K}e^2(k) \tag{9.12}$$

其中 $m=n/K=1,2,\ldots$，平均区间 K 可以近似选为 $K=10N$。步长参数△的选择对 LMS 算法收敛速度的影响可通过监测 ASE(m)来观察。

除了程序的主要部分，也应额外包括计算未知系统的单位脉冲响应，这用单位脉冲序列激励未知系统就可得到。将这个实际的单位脉冲响应与 LMS 算法收敛后的 FIR 模型的单位脉冲响应相比较，为此可以画出二者的曲线图。

宽带信号中窄带干扰的消除

假设有一个信号序列 $\{x(n)\}$，由预期宽带信号序列（$\{w(n)\}$）与一加性窄带干扰序列 $\{s(n)\}$ 之和构成，这两个序列互不相关。这种问题出现在数据通信和信号检测中，预期信号 $\{w(n)\}$ 是扩展频谱信号，而窄带干扰表示来自另一频带用户的信号，或者是来自企图破坏通信或检测系统的干扰台的故意干扰信号。

从滤波观点考虑，我们的目的是设计消除窄带干扰的滤波器。事实上，这种滤波器应在干扰的频带区域设置凹槽。但实际中干扰的频带可能不知道，甚至干扰频带随时间慢变化。

干扰的窄带特性允许我们从序列 $x(n)=s(n)+w(n)$ 过去的样本值估计 $s(n)$，并从

$x(n)$中减去此估值，因为相对$\{w(n)\}$的带宽，$s(n)$的带宽窄得多，所以$s(n)$的样本值间有很强的相关性。换言之，宽带序列$\{w(n)\}$的相关性相对很弱。

干扰对消系统的一般结构如图9.3所示。信号$x(n)$延时D个采样，D要选得足够大以致于宽带信号分量$w(n)$与$w(n-D)$不相关（二者分别包含在$x(n)$和$x(n-D)$中），自适应FIR滤波器的输出为估值：

$$\hat{s}(n)=\sum_{k=0}^{N-1}h(k)x(n-k-D) \tag{9.13}$$

用于优化FIR滤波器系数的误差信号是$e(n)=x(n)-\hat{s}(n)$。使平方误差和最小化又导出了确定最佳系数的线性方程组。由于延时D，迭代调整系数的LMS算法变成

$$h_n(k)=h_{n-1}(k)+\triangle e(n)x(n-k-D),\begin{matrix}k=0,1\dots,N-1\\ n=1,2\dots\end{matrix} \tag{9.14}$$

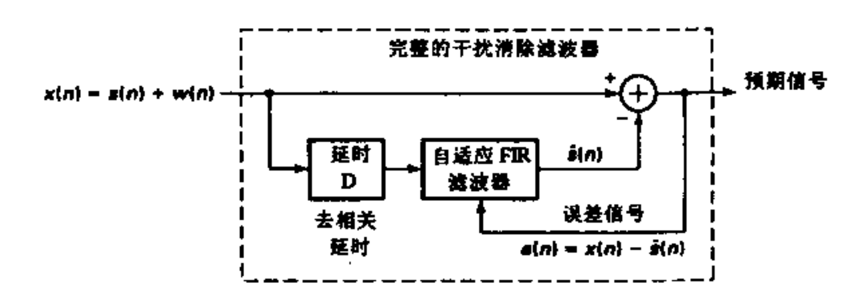


图9.3 估计并对消窄带干扰的自适应滤波器

设计题9.2 正弦干扰对消

实现本课题需要三个基本部分。

1.噪声信号产生器部分：产生零均值随机数宽带序列$\{w(n)\}$，特别是我们可以用前面系统辨识中介绍的rand函数产生均匀分布的随机数序列，该信号功率记为P_w。

2.正弦信号产生器部分：产生一正弦波序列$s(n)=A\sin\omega_0 n$，其中$0<\omega_0<\pi$，A是正弦幅度，正弦序列功率记为P_s。

3.使用lms函数的自适应FIR滤波器程序：该FIR滤波器有N个用LMS算法调整的抽头系数。滤波器长度N作为程序输入变量。

将三个程序块按图9.4组成干扰对消系统，本题中延时$D=1$就够了，因为序列$\{w(n)\}$是白噪声（频谱是平的或不相关）的序列。本题的目的是自适应调整FIR滤波器系数，并研究自适应滤波器特性。

有趣的是将干扰信号选得比所希望的信号$w(n)$强得多，例如$P_s=10P_w$，应注意到LMS算法中选择步长参数所需要的功率$P_x=P_s+P_w$。抽头系数为$\{h(k)\}$的自适应FIR滤波器的频率响应特性$H(e^{j\omega})$在干扰频率上应呈现谐振峰，干扰对消滤波器的频率响应为$H_s(e^{j\omega})=1-H(e^{j\omega})$，所以在干扰频率上应呈现凹槽

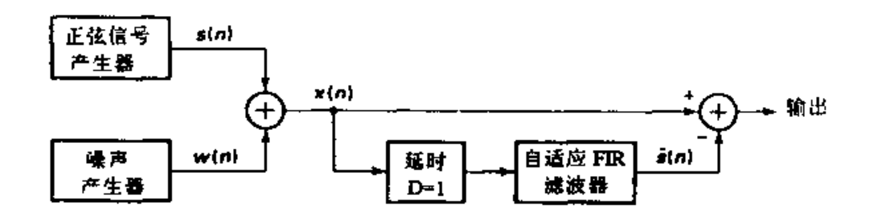


图 9.4　干扰对消举例程序块组成

绘制序列$\{w(n)\}$、$\{s(n)\}$和$\{x(n)\}$的波形图是有趣的。在 LMS 算法收敛后绘制频率响应$H(e^{j\omega})$和$H_s(e^{j\omega})$也很有意思。短时平均平方误差 ASE(m)(由(9.12)定义)可以用来监测 LMS 算法的收敛特性,应当研究自适应滤波器长度对估计质量的影响。

加入不同频率的第二个正弦干扰可使本课题普遍化。假定两频率截然分开,这时$H_s(e^{j\omega})$应呈现两个谐振峰,可以试验滤波器长度对分辨频率很接近的两个正弦信号的影响。

自适应谱线增强

在前节我们介绍过从宽带信号中消除强窄带干扰的一种方法。除目的不同外,自适应谱线增强器(ALE)的构成与图 9.3 所示的干扰对消滤波器相同。

在自适应谱线增强器中,$\{s(n)\}$是预期信号,而$\{w(n)\}$表示掩盖($\{s(n)\}$的宽带噪声。预期信号$\{s(n)\}$可能是单一谱线(一纯正弦)或带宽较窄的信号。通常,宽带信号功率大于窄带信号功率,即$P_w > P_s$,显然,ALE 是一个自调谐滤波器,其频率响应谐振峰在输入正弦频率上或在窄带信号占有的频带上。通过 FIR 窄带滤波方式消除信号频带以外的噪声,因此,相对于$\{w(n)\}$功率而言,信号谱线幅度增强了。

设计题 9.3　自适应谱线增强

本题需要的软件块与干扰对消题中所用的相同,因此上一节给出的说明直接适用于本题,第一个变化是在 ALE 中,条件成为$P_w > P_s$,第二个变化是 ALE 的输出为$\{s(n)\}$,在这些条件下重复上节描述的设计题。

自适应信道均衡

数据通过电话信道的速率通常因引起码间干扰(ISI)的信道失真而受到限制。数据速率低于 2400bits 时,ISI 相对较小,调制解调器工作一般不存在问题。然而,数据速率在 2400bits 以上时,调制解调器中用自适应均衡器来补偿信道失真,所以允许高可靠地进行高速数据传输。在电话信道中,整个系统使用滤波器分离不同频带的信号。这些滤波器引起幅度失真和相位失真,自适应均衡器实质上是一个自适应 FIR 滤波器,其系数用 LMS 算法调整以便校正信道失真。

图 9.5 给出了通过信道传送数据的调制解调器基本组成的方框图。开始,通过传送一短时训练序列来调整均衡器系数,训练序列持续时间一般小于 1 秒,在短的训练周期后,发送机开始传送数据序列$\{a(n)\}$,为了跟踪信道可能发生的慢变化,在接收数据期间必须连续不断地自适应调整均衡器系数。通常,认为判决器输出端的判断是正确的,并用该判决结果代替参

考信号$\{d(n)\}$产生误差信号来完成这种调整，见图 9.5 说明。当判决误码率很低（如小于1%）时，这种方法工作得非常好，偶然的判决错误仅对均衡器系数产生小的调整。

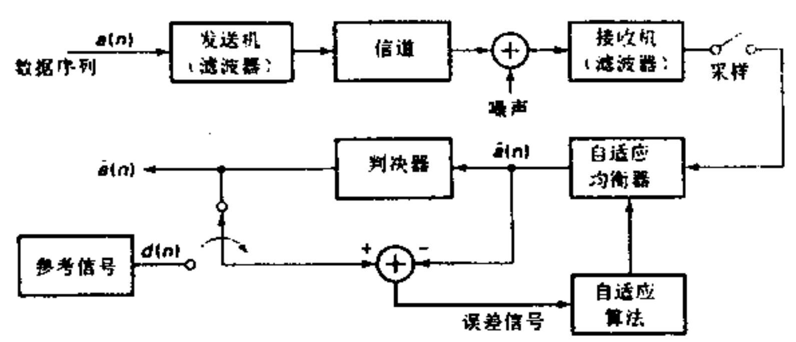


图 9.5　自适应滤波器应用于自适应信道均衡

设计题 9.4　自适应信道均衡

本题的目的是研究在引起码间串扰的信道上传输数据的自适应均衡器的实现，所模拟的系统基本构成如图 9.6。由图可见，需要 5 个基本程序块，请注意到我们省去了电话信道中所要求的载波调制解调，这样做是为了简化模拟程序，但所有处理均涉及复数运算。

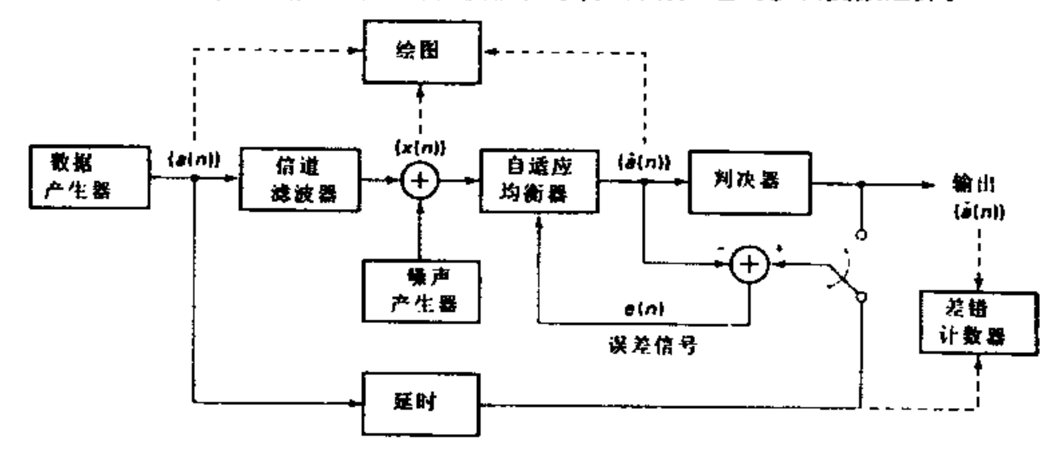


图 9.6　研究自适应均衡器性能的实验

五个部分如下：

1. 数据产生：产生复数信息符号序列$\{a(n)\}$，具体拥有四个等概率符号 $s+js$，$s-js$，$-s+js$以及$-s-js$，其中 s 是比例因子，可设置为 $s=1$ 或作为输入参数。

2. 信道滤波器模块：模拟信道失真的 FIR 滤波器，其抽头系数为$\{c(n),\ 0\leqslant n\leqslant K-1\}$。对无失真传输，设置 $c(0)=1$ 且 $c(n)=0,\ 1\leqslant n\leqslant K-1$。滤波器长度 K 作为输入参数。

3. 噪声产生程序块：用来产生通常在数字通信系统中出现的加性噪声，如果我们要模拟电子器件产生的噪声，则它应满足零均值高斯分布，可用 randu 函数产生。

4. 自适应均衡部分是一个 FIR 滤波器，其抽头系数为$\{h(k),\ 0\leqslant k\leqslant N-1\}$，用 LMS 算法调整这些系数。然而，由于使用复数计算，所以 LMS 算法中的迭代方程变为

$$h_n(k)=h_{n-1}(k)+\triangle e(n)x^*(n-k) \tag{9.15}$$

其中 * 号表示复共轭。

5.判决器模块取出估计值 $\hat{a}(n)$，并按如下判决规律量化为 4 个可能的信号点之一：

$$
\begin{array}{l}
Re[\hat{a}(n)]>0 \quad \text{and} \quad \text{Im} \quad [\hat{a}(n)]>0 \quad \rightarrow \quad 1+j \\
\text{Re}[\hat{a}(n)]>0 \quad \text{and} \quad \text{Im} \quad [\hat{a}(n)]<0 \quad \rightarrow \quad 1-j \\
\text{Re}[\hat{a}(n)]<0 \quad \text{and} \quad \text{Im} \quad [\hat{a}(n)]<0 \quad \rightarrow \quad -1+j \\
\text{Re}[\hat{a}(n)]<0 \quad \text{and} \quad \text{Im} \quad [\hat{a}(n)]<0 \quad \rightarrow \quad -1-j
\end{array}
$$

这种均衡器抑制信道滤波器引入的码间干扰的效果可以通过绘制以下相应序列的二维(实－虚)图形观察到，数据产生器输出$\{a(n)\}$应包含值为 $\pm 1 \pm j$ 的 4 个点，信道失真和加性噪声的影响可通过显示均衡器输入序列$\{x(n)\}$来观察。自适应均衡器的效果可在其系数收敛后，通过绘制它的输出$\{\hat{a}(n)\}$来估计。也可以利用短时均方误差 ASE(n)监测 LMS 算法的收敛特性。请注意，必须在数据产生器输出中引入延时来补偿信道滤波器和自适应均衡器对信号的延时。例如，该延时可设置为最靠近$(N+K)/2$ 的最大整数。最后，用误差计数器来计算所接收数据序列中符号差错的数目，并显示差错数与总符号数之比(差错率)。通过改变 ISI 电平和加性噪声电平可以使差错率变化。

建议对下列三种信道条件进行模拟：

a. 无码间干扰：$c(0)=1, c(n)=0,, 1 \leqslant n \leqslant K-1$

b. 弱码间干扰：$c(0)=1, c(1)=0.2, c(2)=-0.2, c(n)=0, 3 \leqslant n \leqslant K-1$

c. 强码间干扰：$c(0)=1, c(1)=0.5, c(2)=-0.5, c(n)=0, 3 \leqslant n \leqslant K-1$

可将所测量的差错率作为均衡器输入端的信噪比(SNR)的函数进行绘图，其中 SNR 定义为 P_s/P_n，信号功率 $P_s=s^2$，P_n 为噪声产生器输出端的噪声序列功率。

小　结

本章中，我们向读者介绍了自适应 FIR 滤波器及其在系统辨识、干扰对消、窄带频率增强和自适应均衡方面应用的理论和实现，系统阐述了涉及自适应滤波应用的设计题，这些课题都能够用 MATLAB 实现。

第十章　通信中的应用

我们发现 MATLAB 在各种通信系统的模拟中有广阔的应用前景。在本章中，我们将集中讨论与波形表示与编码(特别是语音编码)和数字通信有关的几种应用，尤其是详细叙述连续波形数字化的几种方法及其在语音编码与传输中的具体应用，这些方法是脉冲编码调制(PCM)，差分 PCM 和自适应 PCM(ADPCM)，增量调制(DM)和自适应增量调制(ADM)，以及线性预测编码(LPC)，并提出了用 MATLAB 模拟这些波形编码方法的设计题。

本章最后三个专题讨论在实现数字通信系统接收机时通常会遇到的信号检测应用，对每一个专题，我们介绍一个设计题，其中包含用 MATLAB 模拟检测框图的具体实现方案。

脉冲编码调制(PCM)

PCM 是为了用数字方式传输或存贮信号，对模拟信号进行量化的一种方法。PCM 被广泛应用于电话通信中的语音传送和利用无线电传输的遥测系统中，我们将主要讨论 PCM 在语音信号处理方面的应用。

通过电话线传输的语音信号频带限于 4kHz 以下的频率范围，因此采样这种信号的奈奎斯特速率小于 8kHz(采样数/秒)，每个采样量化为 2^b 个电平之一，并用 b 位二进制序列表示。所以，传输数字化语音信号所要求的速率为每秒 8000b 位。

量化处理的数学模型为

$$\tilde{s}(n)=s(n)+q(n) \tag{10.1}$$

其中 $\tilde{s}(n)$表示 $s(n)$的量化值，$q(n)$表示量化误差，将其看作一加性噪声。假设用均匀的量化器且电平数足够多，则用如下均匀概率密度函数可很好地统计描述量化噪声特性：

$$p(q)=\frac{1}{\Delta},-\frac{\Delta}{2}\leqslant q\leqslant\frac{\Delta}{2} \tag{10.2}$$

其中，量化器的量化步长为 $\Delta=2^{-b}$，量化误差的均方值为：

$$E(q^2)=\frac{\Delta^2}{12}=\frac{2^{-2b}}{12} \tag{10.3}$$

用分贝来度量的噪声均方值为：

$$10\log\left(\frac{\Delta^2}{12}\right)=10\log\left(\frac{2^{-2b}}{12}\right)=-6b-10.8\text{dB} \tag{10.4}$$

可以看出，上述的量化器每增加一位，量化噪声减小 6dB，高质量语音要求每个样本至少量化为 12 位，因此要求速率至少为 96000 位/秒。

语音信号的特性是小幅度比大幅度出现得频繁，然而，均匀量化器在信号的整个动态范围中的量化间隔相同。更好地途径是用非均匀量化器，这种量化器对小的信号幅度提供较密的量化间隔，对大的信号度提供较粗的量化间隔，对 b-位非均匀量化器，其量化误差的均方值小于(10.4)给出的值。为了得到非均匀量化器特性，可先使信号通过压缩幅度的非线性器件，后面再接一均匀量化器。例如，在美国和加拿大的通信系统中使用的对数压缩器(称为 μ-律压缩器)具有如下输入输出幅度特性：

$$y=\frac{\ln(1+\mu|s|)}{\ln(1+\mu)}\text{sgn}(s);\quad |s|\leqslant 1,|y|\leqslant 1 \tag{10.5}$$

其中，s 是归一化输入，y 是归一化输出，sgn(·)是符号函数，μ 是控制压缩特性的参数。

在美国和加拿大的语音波形编码中，选定 $\mu=255$ 为标准值，与均匀量化器相比，该 μ 值的效果是使量化噪声功率降低 24dB。所以，一个与 $\mu=255$ 的对数压缩器连用的 8 位量化器产生的语音质量与无压缩的 12 位量化器相同。因此，压缩了的 PCM 语音信号具有 64000 位/秒的速率。

在欧洲的通信系统中所用的对数压缩标准为 A-律，定义如下：

$$y=\begin{cases}\dfrac{1+\ln(A|s|)}{1+\ln A}\text{sgn}(s), & \dfrac{1}{A}\leqslant|s|\leqslant 1\\[2ex] \dfrac{A|s|}{1+\ln A}\text{sgn}(s), & 0\leqslant|s|\leqslant\dfrac{1}{A}\end{cases} \tag{10.6}$$

其中，A 选为 87.56。虽然(10.5)与(10.6)是不同的非线性函数，但两种压缩特性非常相似。图 10.1 给出了这两种压缩函数，请注意二者很强的相似性.

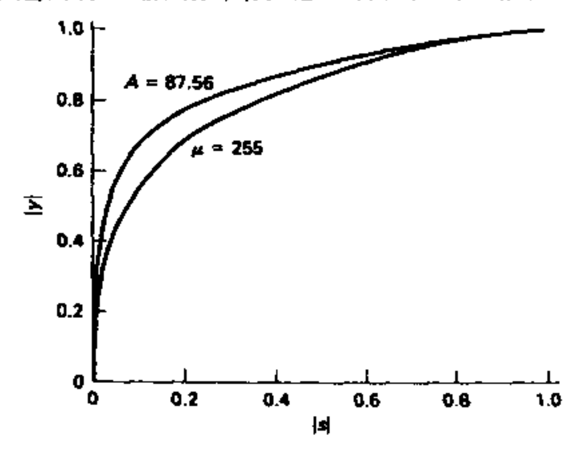


图 10.1 μ-律与 A-律非线性的比较

由量化值恢复信号时，解码器用逆对数关系来扩展信号幅度，例如，在 μ-律中，逆关系如下：

$$|s| = \frac{(1+\mu)^{|y|} - 1}{\mu}; \ |y| \leqslant 1, \ |s| \leqslant 1 \tag{10.7}$$

组合的一对压缩-扩张器称为压扩器。

设计题 10.1 PCM

本课题的目的是理解 PCM 压缩(线性到对数)与 PCM 扩张(对数到线性)，对本题写出如下三个 MATLAB 函数。

1. 实现(10.5)的 μ-律压缩函数，它接受一零均值归一化($|s| \leqslant 1$)信号，产生压缩后的零均值归一化信号，μ 作为可以选定的自由参数。

2. 量化函数：它接受零均值输入，产生一个 b-位量化的整数输出，b 可以任意选定。

3. 实现(10.7)的 μ-律扩张器函数：接收整数输入，产生参数 μ 值可变的零均值输出。

为了进行仿真，产生如下序列的大量(10000 或更多)样本：(a)锯齿形序列；(b)指数脉冲序列；(c)正弦序列；(d)方差较小的随机序列。须注意取其归一化频率为无理数，使产生非周期序列(即样本不会重复)。如，可以用下式产生正弦序列：

$$s(n) = 0.5\sin(n/33), 0 \leqslant n \leqslant 10{,}000$$

由第二章讨论可知，该序列是非周期的，而其包络是周期的，其他序列也可用类似方式产生，将这些信号通过上述 μ-律压缩器、量化器和扩张器函数进行处理(如图 10.2 所示)，计算信号对量化噪声之比(SQNK)的 dB 数

$$\mathrm{SQNK} = 10\log_{10}\left(\frac{\sum_{n=1}^{N} s^2(n)}{\sum_{n=1}^{N}(s(n) - s_q(n))^2}\right)$$

对不同的 b-位量化器，分别确定使 SQNK 最大的 μ 值，绘出输入和输出波形，并讨论其结果。

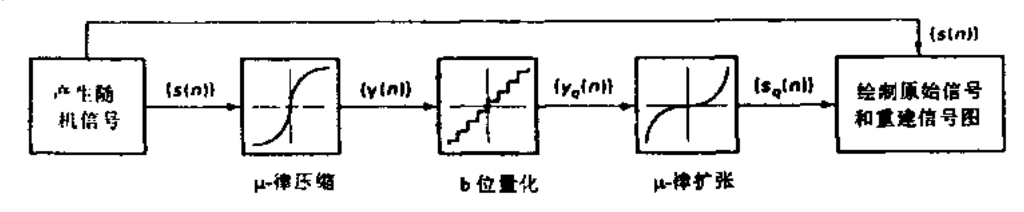


图 10.2 PCM 方案

差分 PCM(DPCM)

在 PCM 中,波形的每个样本独立进行编码。然而,以奈奎斯特速率或更高速率采样的绝大多数信号(包括语音信号),其相邻的样本之间呈现明显的相关性,换言之,相邻采样幅度间的平均变化相对较小。所以,利用采样中多余度的编码方案将使语音信号的码率降低。

一种相对简单的解决方法是对相邻样本之差编码而不是对样本本身编码,由于相邻样本之差比实际样本幅度小,所以表示差信号需要较小的位数。这种普通方法的一种改进方案是用前面 P 个样本预测当前样本,为了明确起见,设 $s(n)$ 表示语音信号的当前样本,而 $\hat{s}(n)$ 表示 $s(n)$ 的预测值,其定义为:

$$\hat{s}(n)=\sum_{i=1}^{P}a(i)s(n-i) \tag{10.8}$$

所以,$\hat{s}(n)$ 是过去 P 个采样的加权线性组合,而 $a(i)$ 是预测器(滤波器)系数,选择 $a(i)$ 以便使 $s(n)$ 与 $\hat{s}(n)$ 之差的某种函数最小化。

一种在数学上和实际中都较方便的误差函数是平方误差和,以此作为预测器的实现准则,选取 $a(i)$ 使之最小化:

$$\varepsilon_p \triangleq \sum_{n=1}^{N}e^2(n)=\sum_{n=1}^{N}\Big[s(n)-\sum_{i=1}^{P}a(i)s(n-i)\Big]^2 \tag{10.9}$$

$$=r_{ss}(0)-2\sum_{i=1}^{P}a(i)r_{ss}(i)+\sum_{i=1}^{P}\sum_{j=1}^{P}a(i)a(j)r_{ss}(i-j)$$

其中,$r_{ss}(m)$ 是采样序列 $s(n)$ 的自相关函数,定义为:

$$r_{ss}(m)=\sum_{i=1}^{N}s(i)s(i+m) \tag{10.10}$$

使 ε_p 关于预测器系数 $\{a_i(n)\}$ 最小化,可导出线性方程组(称为正交方程):

$$\sum_{i=1}^{P}a(i)r_{ss}(i-j)=r_{ss}(j),\ j=1,2,\cdots,p \tag{10.11}$$

或以矩阵形式

$$\mathbf{Ra}=\mathbf{r}\Rightarrow\mathbf{a}=\mathbf{R}^{-1}\mathbf{r} \tag{10.12}$$

其中 $\mathbf{R}$ 为自相关矩阵,$\mathbf{a}$ 为系数矢量,$\mathbf{r}$ 为自相关矢量,由此可确定预测器系数的值。

上面叙述了确定预测器系数的方法,现在来考虑实际 DPCM 系统方框图,如图 10.3 所示。该图中,预测器用跨接量化器的反馈环实现,预测器的输入记为 $\tilde{s}(n)$,它表示信号样本

$s(n)$经量化处理过的值,预测器输出为:

$$\hat{s} = \sum_{i=1}^{N} a(i)\tilde{s}(n-i) \tag{10.13}$$

差值

$$e(n) = s(n) - \hat{s}(n) \tag{10.14}$$

是量化器的输入,$\tilde{e}(n)$表示其输出、量化后的预测误差 $\tilde{e}(n)$ 的每个值编码为一个二进制数序列,并通过信道传送到接收器,量化误差 $\tilde{e}(n)$ 也加到预测值 $\hat{s}(n)$ 上以得到 $\tilde{s}(n)$ 。

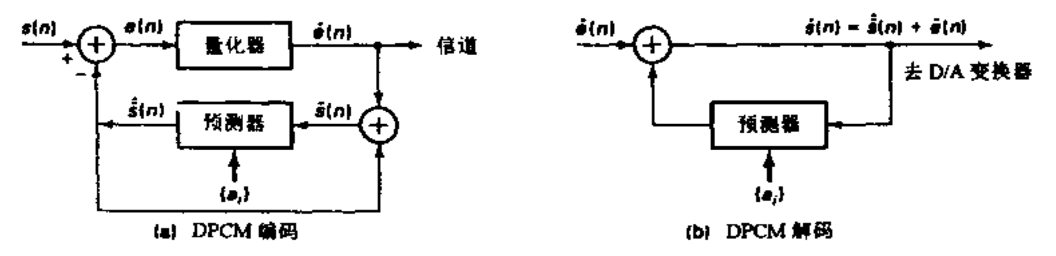


图 10.3 DPCM 编码器的框图 (a)编码器 (b)解码器

在接收端,组成与发送端相同的预测器,其输出 $\hat{s}(n)$加到 $\tilde{e}(n)$得到 $\tilde{s}(n)$。信号 $\tilde{s}(n)$ 是预测器要求的激励信号,也是预期的输出序列,通过滤波该序列,得到重构信号 $\tilde{s}(t)$,如图 10.3.b 所示。

使用上述跨接量化器的反馈环保证了 $\tilde{s}(n)$中的误差仅是量化误差 $q(n) = \tilde{e}(n) - e(n)$,且在译码器的实现中没有以前量化误差的积累,即:

$$q(n) = \tilde{e}(n) - e(n) = \tilde{e}(n) - s(n) + \hat{s}(n) = \tilde{s}(n) - s(n) \tag{10.15}$$

因此, $\tilde{s}(n) = s(n) + q(n)$ 。这意味着量化后的样本 $\tilde{s}(n)$与输入 $s(n)$之差为量化误差 $q(n)$, 与所用的预测器无关,所以量化误差不会积累。

在图 10.3 给出的 DPCM 系统中,信号样本 $s(n)$的估计(或预测)值 $\hat{s}(n)$是通过取过去的估计值($\tilde{s}(n-k), k = 1,2,\ldots,p$) 的线性组合得到的,如(10.13)所述.加入量化误差的过去值可使估计质量得到改善,明确地讲,$s(n)$的估计表示为:

$$\hat{s}(n) = \sum_{i=1}^{p} a(i)\tilde{s}(n-i) + \sum_{i=1}^{m} b(i)\tilde{e}(n-i) \tag{10.16}$$

其中 $b(i)$是对 $\tilde{e}(n)$进行滤波的滤波器系数,发送端编码器和接收端译码器的框图如图 10.4 所示。选择两组系数 $a(i)$和 $b(i)$使误差 $e(n) = \tilde{s}(n) - s(n)$ 的某一函数(如平方误差和)最小化。

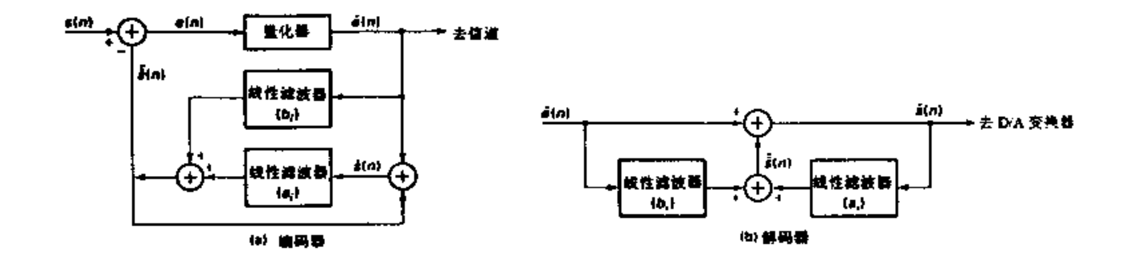


图 10.4 用线性滤波误差序列修正的 DPCM

如果对误差序列 $e(n)$用对数压缩的4-位量化,DPCM 可达到32000位/秒的高质量效果，它是对数压缩 PCM 的1/2。

设计题 10.2 DPCM

本题目的目的是理解 DPCM 编码运算过程,为了进行仿真,用如下形式的极-零点信号模型产生有关的随机序列

$$s(n)=a(1)s(n-1)+b_0x(n)+b_1x(n-1) \tag{10.17}$$

其中,$x(n)$是零均值单位方差的序列,可用 filter 函数来产生,也可用设计题 10.1 中开发的序列进行仿真,对本课题开发如下三个 MATLAB 程序块。

1. 给定输入信号 $s(n)$,实现(10.12)的典型预测器函数。

2. 实现图 10.3a 所示框图的 DPCM 编码器函数,该函数接受一个零均值输入序列,产生量化后的 b-位整数误差序列,b是可改变的参数。

3.图 10.3b 的 DPCM 译码函数,该函数从量化的误差序列重构信号。

给定信号,对几个 p 阶预测模型进行实验,并求出最佳阶数。将这种 DPCM 与设计题10.1的 PCM 系统进行比较,评价实验结果。扩展此方案,使之包含(10.16)描述的 m 阶滑动平均滤波器的系统。

自适应 PCM(ADPCM)与 DPCM

通常,语音信号功率随时间缓慢变化,然而, PCM 和 DPCM 编码器是以语音信号功率不变化且量化器满载为条件设计的。若使它们适应语音功率电平随时间的缓慢变化,则可使其效果和性能得到改进。

在 PCM 和 DPCM 中,由均匀量化器对功率慢变的输入信号进行量化所产生的量化误差的方差(量化噪声功率)将是时变的。降低量化噪声动态范围的一种改进方法是使用自适应量化器。

自适应量化器分为前馈式或反馈式,前馈自适应量化器根据输入语音信号方差(功率)的测量对每个信号样本调整其量化阶距大小。例如,以滑动窗估计器的估计方差为:

$$\hat{\sigma}_{n+1}^2=\frac{1}{M}\sum_{k=n+1-M}^{n+1}s^2(k) \tag{10.18}$$

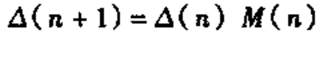
则量化器的量化阶距为：

$$\Delta(n+1)=\Delta(n)\delta_{n+1} \tag{10.19}$$

在这种情况下，必须将 $\Delta(n+1)$传送到译码器以便恢复信号，自适应量化器利用量化器输出调整量化阶距大小。我们可以取量化阶距为

$$\Delta(n+1)=\alpha(n)\ \Delta(n) \tag{10.20}$$

其中，比例因子 $\alpha(n)$取决于以前的量化器输出。例如，如果前面的输出小，我们可选择 $\alpha(n)<1$，以便提供更细的量化，反之，如果前面的输出大，则应当增大量化阶距，来减小信号被限幅的可能性。这种算法已成功地用于语音信号编码。图 10.5 举例说明了这种(3-位)量化器，在其中根据如下关系递归调整量化阶距



$$\Delta(n+1)=\Delta(n)\ M(n)$$

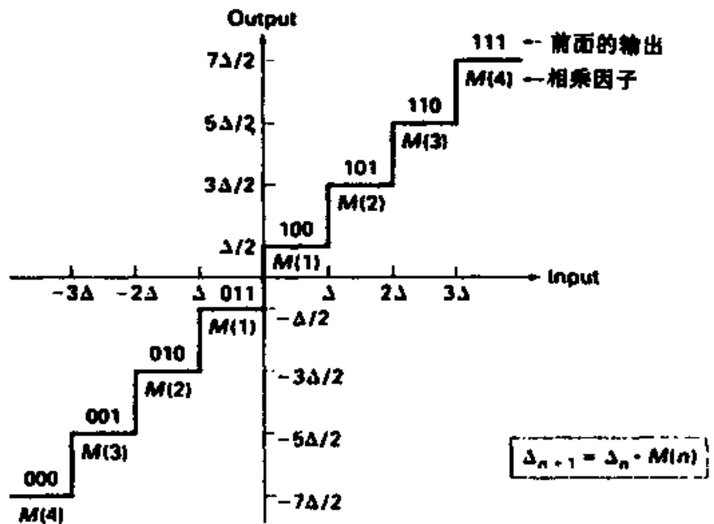


图 10.5 具有自适应量化阶距的量化器举例

其中 $M(n)$为相乘因子，其值取决于采样 $s(n)$的量化电平，$\Delta(n)$是量化器处理 $s(n)$的量化阶距。在文献[13]中给出优化的语音编码相乘因子值，对 2-、3-和 4-位 PCM 和 DPCM 的 $M(n)$值见表 10.1。

表 10.1 自适应量化阶距调整的相乘因子[10]

	PCM			DPCM		
	2	3	4	2	3	4
M(1)	0.60	0.85	0.80	0.80	0.90	0.90
M(2)	2.20	1.00	0.80	1.60	0.90	0.90
M(3)		1.00	0.80		1.25	0.90
M(4)		1.50	0.80		1.70	0.90
M(5)			0.80			1.20
M(6)			0.80			1.60
M(7)			0.80			2.00
M(8)			0.80			2.40

在 DPCM 中，也可将预测器做成自适应的，于是 ADPCM 中预测器系数也周期地改变来反映变化的语音信号的统计特性。公式(10.11)给出的线性方程组仍然适用，但 $s(n)$ 的短期自相关函数 $r_{ss}(m)$ 将随时间变化。

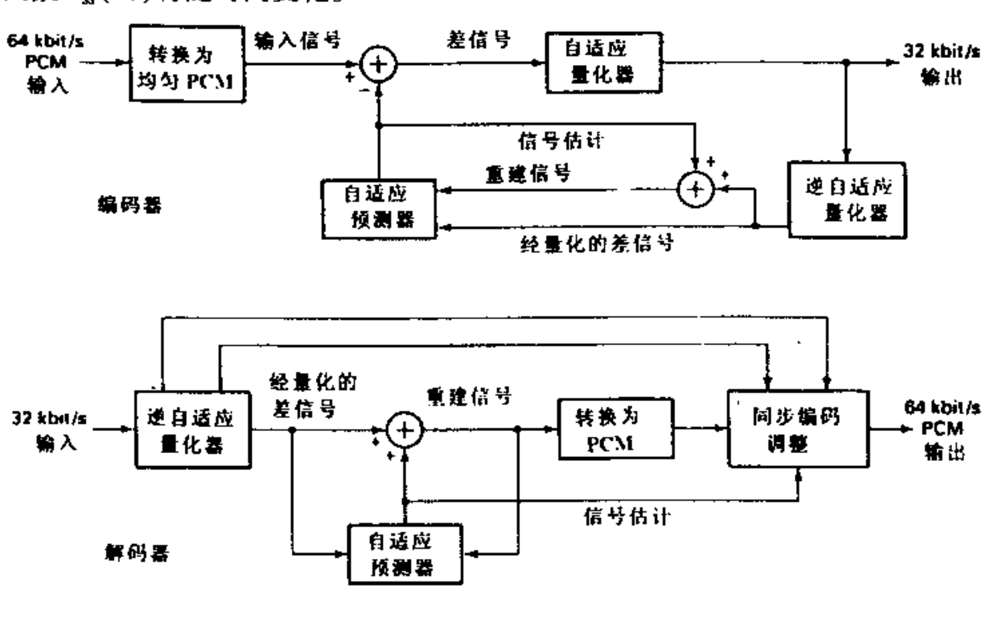


图 10.6 ADPCM 方框图

ADPCM 标准

图 10.6 以方框图形式说明一个 32000 位/秒的 ADPCM 编码器和译码器，它已被认定为语音信号在电话线路上传输的(CCITT)国际标准。ADPCM 编码器是设计来接收 64000 位/秒的 8-位 PCM 压缩采样信号，利用自适应预测和自适应 4-位量化器将信道位率降到 32000 位/秒。ADPCM 解码器接收这个 32000 位/秒的数据流并将它重构为 8-位压缩 PCM 信号。由此得到图 10.7 所示的结构，其中，ADPCM 编/译码器被嵌入 PCM 系统。虽然这种 ADPCM 编/译码器可直接用于语音信号，但实际中为了与已广泛用在电话网中的 PCM 系统兼容，它与 PCM 系统的接口是必须的。

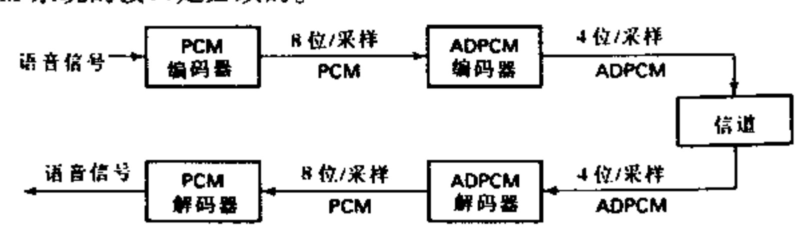


图 10.7 ADPCM 与 PCM 系统的连接

ADPCM 编码器接收 8-位 PCM 压缩信号并将其扩展到每样本 14-位的线性数据以便处理。从该 14-位线性值中减去预测值，产生的差信号样本供给量化器，再对差信号进行自适应量化，产生通过信道传输的 4-位输出。

编码器和译码器根据其产生的 ADPCM 值不断修正其内部参数。因此，将一个包含逆自适应量化器的 ADPCM 译码器嵌入编码器，以使根据同样的数据不断修正内部所有参数，这就

保证了编码器和译码器同步工作，而不需要传送任何有关内部参数值的信息。

自适应预测器计算前面 6 个解量化(dequantized)差值的加权平均和末 2 个预测值。所以，这种预测器根本上是由(10.16)给出的差分方程决定的滤波器，它有 2 个极点($p=2$)和 6 个零点($m=6$)，其滤波系数对每个输入样本不断地自动作调整。

在接收译码器和嵌入在编码器中的译码器中，用传来的 4－位 ADPCM 值不断调整逆自适应量化器。它的输出是经解量化后的差值信号，该解量化值与自适应预测器的输出值相加得到重构的语音样本。这个信号就是译码器的输出，接收机中再将其转换成压缩 PCM 形式。

设计题 10.3　ADPCM

本题的任务是熟悉并理解 ADPCM 及其与 PCM 编/译码器的接口。如上所述，ADPCM 编/译码器按图 10.7 所示的那样插入在 PCM 压缩器与 PCM 扩张器之间。现要将已开发的 MATLAB 的 PCM 和 DPCM 程序用于本题。

像 PCM 题那样，PCM－ADPCM 编/译码系统的输入可由内部产生的波形数据文件提供，编/译码器输出可以绘成图。在用 PCM 系统输出信号(设计题 10.1)作为输入，和用原来的输入信号两种情况下，要求对 PCM－ADPCM 编/译码器输出进行比较。

增量调制(DM)

增量调制可看作 DPCM 的简化形式，它用两个电平(1－位)量化器与固定的一阶预测器联用，DM 编/译码器的框图见图 10.8。注意到：

$$\hat{s}(n)=\tilde{s}(n-1)=\hat{s}(n-1)+\tilde{e}(n-1) \tag{10.21}$$

因为：

$$q(n)=\tilde{e}(n)-e(n)=\tilde{e}(n)-[s(n)-\hat{s}(n)]$$

从而得到：

$$\hat{s}(n)=s(n-1)+q(n-1) \tag{10.22}$$

因此，$s(n)$的估计(预测)值实际上是用量化噪声 $q(n-1)$修正前一个采样值 $s(n-1)$得到的。我们还注意到差分方程(10.21)表示一个输入为 $\tilde{e}(n)$ 的积分器。因此，一步预测器的一种等效实现是输入为量化后的差值信号 $\tilde{e}(n)$ 的累加器。通常，量化后的差值信号用某个值(如 Δ_1)来度量，这个值称为量化台阶，图 10.9 给出这种等效实现。实质上，图 10.9 所示的编码器用线性阶梯函数近似波形 $s(t)$，为了逼近得好些，波形 $s(t)$相对采样速率而言，必须是慢变化的。这种要求意味着采样速率必须是奈奎斯特频率的几倍(至少 5 倍)，通常在译码器中接入低通滤波器来平滑重构信号中的不连续点。

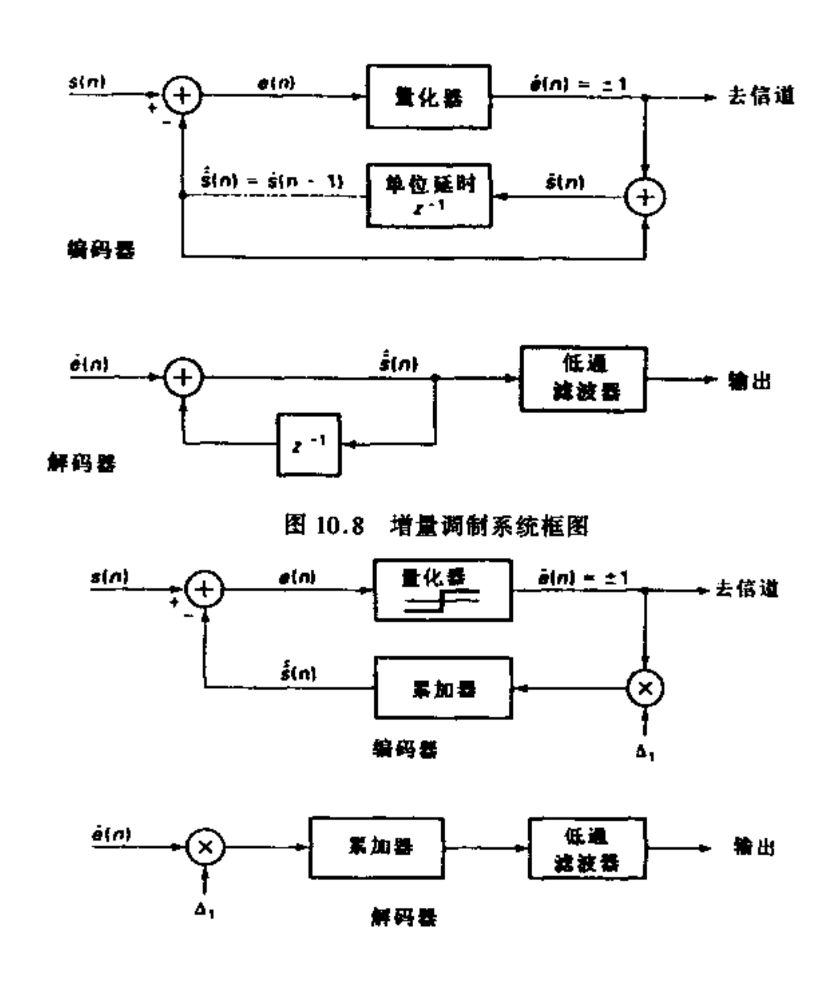


图 10.8 增量调制系统框图

图 10.9 增量调制系统的等效实现

自适应增量调制(ADM)

对任一给定的采样速率,DM 编码器的性能受到图 10.10 所示的两种类型的失真的限制,一类称为斜率-过载失真,它是由于使用太小的量化台阶以至于跟随不上波形中斜率较大的部分;第二类失真(称为颗粒噪声)是由于在波形中斜率小的部分使用太大的量化台阶产生的。要同时使这两类失真最小就会导致对量化台阶的矛盾要求。

另一种可取的解决办法是采用适合源信号短期特性的可变量化台阶,即当波形斜率大时量化台阶增大,反之量化台阶减小。

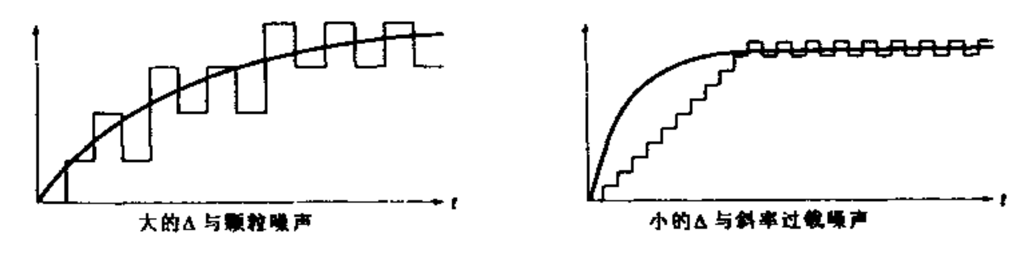


图 10.10 DM 编码器中两类失真

我们可以用多种方法来在每一次迭代中自适应地设定量化台阶,量化后的差值序列能较

好地提供被编码波形的斜率信息。当 $\tilde{e}(n)$ 在相邻两次迭代之间变符号时,就表示当前波形斜率相对较小。反之,当波形有陡峭的斜度时,相邻 $\tilde{e}(n)$ 值将有相同的符号。注意到这些,就有可能设计出根据 $\tilde{e}(n)$ 的相邻值减小或增大量化台阶的算法。[12]提出了一种比较简单的规则,它按照如下关系自适应地修改量化台阶。

$$\Delta(n)=\Delta(n-1)K^{\tilde{e}(n)\tilde{e}(n-1)},n=1,2,... \tag{10.23}$$

其中 $K\geqslant1$ 是一个常数,选取其值使总失真最小。一种采用该自适应算法的 DM 编/译码器的框图如图 10.11 所示。

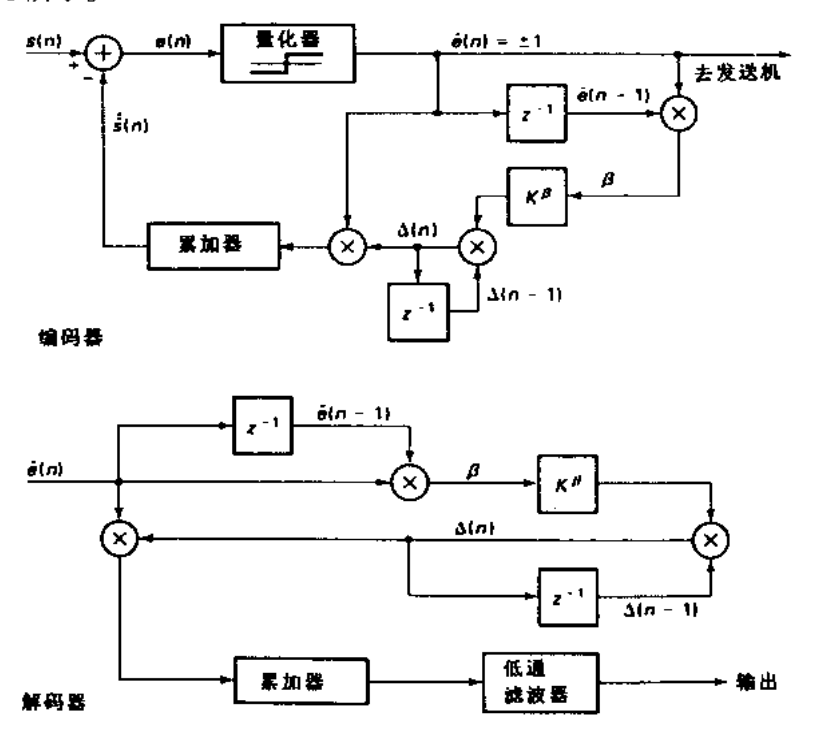


图 10.11 具有自适应量化台阶的增量调制系统举例

几种其他方式的自适应 DM 编码方案在科技文献中已有研究和介绍。首先由[8]提出的一种特别有效且流行的技术被称为连续变化斜率增量调制(CVSD),在 CVSD 中,自适应量化台阶参数可表示为:如果 $\tilde{e}(n)$,$\tilde{e}(n-1)$ 和 $\tilde{e}(n-2)$ 符号相同,则:

$$\Delta(n)=\alpha\Delta(n-1)+k_1 \tag{10.24}$$

否则

$$\Delta(n)=\alpha\Delta(n-1)+k_2 \tag{10.25}$$

参数 α, k_1 和 k_2 的选择应满足:$0<\alpha<1$, 且 $k_1>k_2>0$ 。有关这种和其他各种自适应 DM 的进一步讨论,有兴趣的读者可参阅文献[13]和[6]以及这些文献中包含的广泛的参考资料。

设计题 10.4　增量调制和自适应增量调制

本题旨在理解对波形的 DM 和 ADM 编码，本题包括编写图 10.9 所示的 DM 和图 10.11 所示的 ADM 编/译码器的 MATLAB 函数，解码器中的低通滤波器可用线性相位 FIR 滤波器实现。例如，单位脉冲响应如下式的汉宁(Hanning)滤波器。

$$h(n)=\frac{1}{2}\left[1-\cos\left(\frac{2\pi n}{N-1}\right)\right],0\leqslant n\leqslant N-1 \tag{10.26}$$

其中长度 N 的选择范围为 $5\leqslant N\leqslant 15$。

DM 和 ADM 系统的输入可由题 10.1 中产生的波形提供，但应把采样频率提高 5～10 倍。绘制解码器输出的图形，并对(来自 DM 和 ADM 解码器的)输出信号和原输入信号作比较。

语音信号的线性预测编码(LPC)

用于语音分析与合成的线性预测编码(LPC)方法基于将声道建模为线性全极点(IIR)滤波器，其系统函数为

$$H(z)=\frac{G}{1+\sum_{k=1}^{P}a_p(k)z^{-k}} \tag{10.27}$$

其中，P 是极点数目，G 是滤波器增益，而$|a_p(k)|$是决定这些极点的参数。

有两种不相容的模拟浊音和清音语声的激励函数，在短时基上，浊音是周期的，基频为 F_0 (或音调周期为 $1/F_0$)，F_0 与讲话者有关。所以，用周期等于预期音调的脉冲激励全极点滤波器模型产生浊音，用随机噪声发生器的输出激励全极点滤波器模型产生清音，这种模型如图 10.12所示。

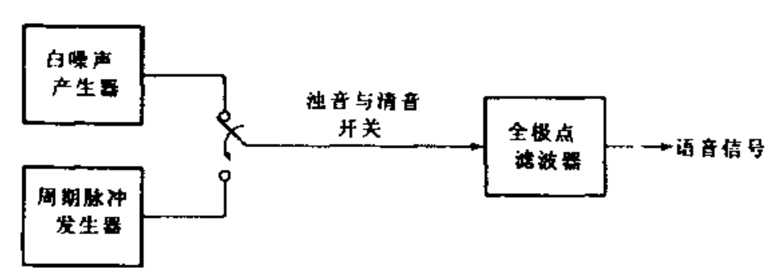


图 10.12　语音信号产生模型框图

给定语音信号的短时部分(一般约为 20ms 或以 8kHz 采样的 160 个样本)，则发送机中的语音编码器必须确定适当的激励函数、浊音语调周期、增益参数 G 以及系数 $a_p(k)$，说明这种语音编码系统的方框图见图 10.13。模型参数由采样数据自适应地确定，并编码为二进制序列传送到接收机，在接收端由模型和激励信号合成语音信号。

由语音样本利用线性预测法容易确定全极点滤波器模型的参数，具体地讲，FIR 线性预测器的输出为

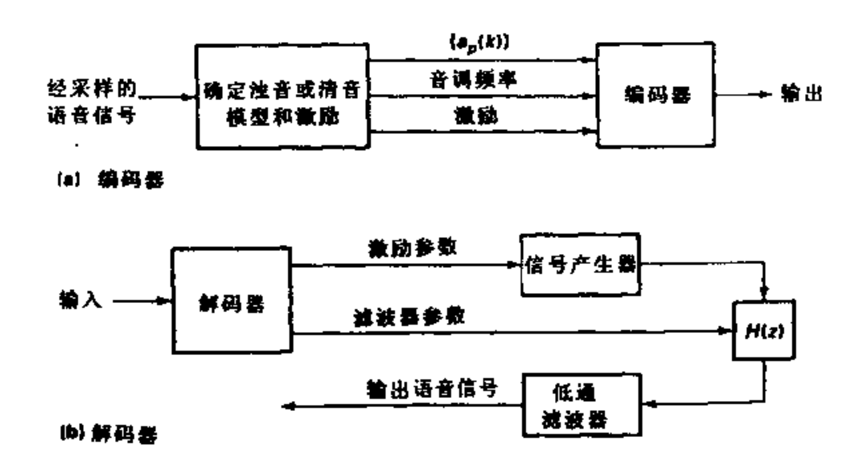


图 10.13 LPC 编码器与解码器

$$\hat{s}(n) = -\sum_{k=1}^{p} a_p(k)s(n-k) \tag{10.28}$$

相应的观测采样 $s(n)$与预测值 $\hat{s}(n)$之间的误差为

$$e(n) = s(n) + \sum_{k=1}^{p} a_p(k)s(n-k) \tag{10.29}$$

使平方误差和最小化，即

$$\varepsilon = \sum_{n=1}^{N} e^2(n) = \sum_{n=1}^{N} \left[s(n) + \sum_{k=1}^{p} a_p(k)s(n-k) \right]^2 \tag{10.30}$$

我们就能够确定模型的极点参数$\{a_p(k)\}$，将 ε 对各个参数求导并使之为零，则得到 p 个线性方程：

$$\sum_{k=1}^{p} a_p(k) r_{ss}(m-k) = -r_{ss}(m),\ m = 1,2\ldots,p \tag{10.31}$$

其中 $r_{ss}(m)$为序列 $s(n)$的自相关，定义为：

$$r_{ss}(m) = \sum_{n=1}^{N} s(n)s(n+m) \tag{10.32}$$

线性方程组(10.31)可以用矩阵形式表示为：

$$\mathrm{R}_{ss}a = -\mathrm{r}_{ss} \tag{10.33}$$

其中 $\mathbf{R}_{ss}$是 $p \times p$ 自相关矩阵,$\mathbf{r}_{ss}$是 $p \times 1$ 自相关向量,a 是 $p \times 1$ 模型参数向量,因此

$$a = -\mathbf{R}_{ss}^{-1}\mathbf{r}_{ss} \tag{10.34}$$

这些方程也可以用 Levinson – Durbin 算法[19]最有效地进行递归求解,而不用求矩阵的逆。然而,用 MATLAB 时矩阵求逆是方便的。用第六章中开发的 MATLAB 函数 dir2latc 可以将全极点滤波器参数$\{a_p(k)\}$ 变换到全极点格型参数$\{K_i\}$(称为反射系数)。

注意到滤波器的输入-输出方程

$$s(n) = -\sum_{k=1}^{p} a_p(k)s(n-k) + Gx(n) \tag{10.35}$$

其中 $x(n)$是输入序列,就可以得到滤波器增益参数,显然,

$$Gx(n) = s(n) + \sum_{k=1}^{p} a_p(k)s(n-k) = e(n)$$

所以

$$G^2 \sum_{n=0}^{N-1} x^2(n) = \sum_{n=1}^{N-1} e^2(n) \tag{10.36}$$

如果设计的输入激励归一化为单位能量,则

$$G^2 = \sum_{n=0}^{N-1} e^2(n) = r_{ss}(0) + \sum_{k=1}^{p} a_p(k) r_{ss}(k) \tag{10.37}$$

因此, G^2 设置为等于按最小平方最优化所得残留误差的能量。

一旦计算出 LPC 系数,就能确定输入语音帧是不是浊音。如果是浊音,还能确定音调,此任务通过计算如下等式完成

$$r_e(n) = \sum_{k=1}^{p} r_a(k) r_{ss}(n-k) \tag{10.38}$$

其中 $r_a(k)$定义为预测系数的自相关序列

$$r_a(k) = \sum_{i=1}^{p} a_p(i)a_p(i+k) \tag{10.39}$$

在 20ms 采样帧中,通过在 3 ~ 5ms 的时间区间探测归一化序列 $r_e(n)/r_e(0)$ 的峰值来检测音

调，如果测得峰值大于等于 0.25，则认为该语音帧为音调周期等于 N_p 的浊音，这里 $n=N_p$ 满足 $r_e(N_p)/r_e(0)$ 为最大值，如果峰值小于 0.25，则认为该语音帧为清音，且音调为零。

LPC 系数值，音调周期以及激励类型被传送到接收机，接收机中使合适的激励通过全极点声道模型来合成语音信号。一般，音调周期需要 6 位，对增益的动态范围进行对数压缩后用 5 位表示，对预测系数编码时，为了精确表示，每个系数需要 8～10 位。精度如此高的原因是较小的预测系数变化会导致滤波器模型极点位置较大变化。通过传送反射系数 $\{K_i\}$ 可使精度要求降低，K_i 有较小的动态范围，即 $|K_i|<1$，每个系数 6 位就足够了。因此，对 10 阶预测器，每帧分配给模型参数的总位数是 72，如果模型参数每 20ms 改变一次，要求的位率为 3600 位/秒，因为通常是将反射系数传送到接收机的，所以接收端的合成滤波器用第 6 章中介绍的全极点格型滤波器实现。

设计题 10.5　线性预测编码(LPC)

本题的目的是通过 LPC 编码器把语音信号分解，然后通过相应的 LPC 解码器将其合成，使用几个 MATLAB 已为此提供的后缀为 wav 的语声文件(采样速率 8000 次/秒)．将语音信号分解成短时段(长度为 120～150 个样本)，并对每一段进行处理以便确定合适的激励函数(浊音或清音)、浊音的音调周期、系数 $\{a_p(k)\}(p\leqslant 10)$ 和增益 G，实现合成的解码器是一个全极点格型滤波器。本方案的输出是可与原语音信号相比较的合成语音信号，要求定性评价由于分解/合成引起的失真效应。

双音多频(DTMF)信号

DTMF 是按键电话通信的德国名称，它等效于贝尔系统中所用的接触音频系统，DTMF 也广泛应用于电子邮件和银行系统中，这些系统中用户可从电话发送 DTMF 信号来选择菜单进行操作。

在 DTMF 通信系统中，高频音与低频音的一个组合表示一个特定的数字或字符 * 和 #。8 个频率按图 10.14 分配共提供 16 个字符，其中 12 个分配如图所示，而其他 4 个保留以备进一步扩展用。

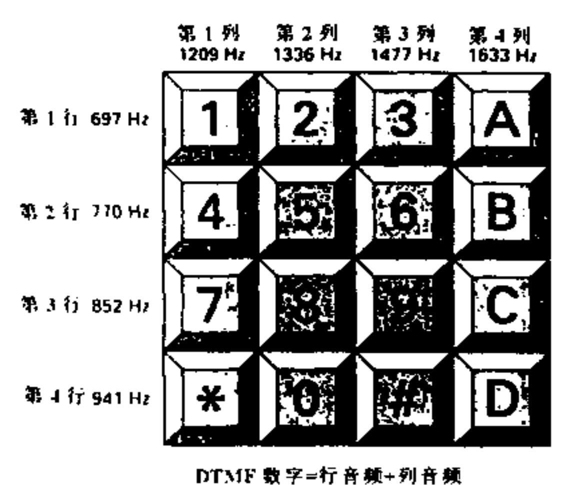


图 10.14　DTMF 数字

DTMF 信号容易用软件产生并用调谐在这 8 个音频的数字滤波器检测，也可用软件实现检测。通常，借助编译码器芯片或线性 A/D 和 D/A 变换器将 DTMF 信号接到模拟域，编译码器包含作为双向模/数接口所必须的 A/D 和 D/A，采样和滤波电路。

DTMF 音频可以用数学方法或查表法产生。在硬件实现(即用数字处理器)中，两个正弦波的数字样本用数学方式产生，并按比例相加在一起，将和信号进行对数压缩并送到编译码器变换成模拟信号。在 8kHz 的采样频率上，硬件必须每 125 毫秒输出一个样本，在这种情况下，正弦波形的值可以迅速地计算出来，可节省查表法所要求的大容量存贮空间，进行仿真试验时，MATLAB 查表法可能是一种好途径。

在接收端，收到的对数压缩的 8-位数据码，经对数扩张为 16 - 位线性码形式，然后检测音频频率来确定所发送的数字。检测算法可以是用 FFT 算法的 DFT，或是用一组滤波器实现。当要检测的音频数目比较少时，滤波器组实现更有效。下面要叙述如何用 Goertzel 算法实现 8 个调谐滤波器。

回想在第五章讨论的 N 点数据序列$\{x(n)\}$的 DFT 为：

$$X(k)=\sum_{n=0}^{N-1}x(n)W_N^{nk},k=0,1,\ldots,N-1 \tag{10.40}$$

如果用 FFT 算法实现该 DFT 计算，计算量(复数乘法和加法)是 $N\log_2 N$，这样我们立即得到 DFT 的所有 N 个值，然而，如果希望计算 DFT 的 M 点而 $M<\log_2 N$，则直接计算 DFT 更有效。下面介绍的 Goertzel 算法根本上是计算 DFT 的一种线性滤波方法，它提供了直接计算之外的另一种方法。

Goertzel 算法

Goertzel 算法利用相位因子$\{W_N^k\}$的周期性，并使我们能将 DFT 运算表示为线性滤波运算，由于 $W_N^{-kN}=1$，我们可用该因子去乘 DFT，则

$$X(k)=W_N^{-kN}X(k)=\sum_{m=0}^{N-1}x(m)W_N^{-k(N-m)} \tag{10.41}$$

注意到，(10.41)就是卷积形式。的确，如果定义序列 $y_k(n)$为

$$y_k(n)=\sum_{m=0}^{N-1}x(m)W_N^{-k(n-m)} \tag{10.42}$$

则 $y_k(n)$显然就是长度为 N 的有限长输入序列 $x(n)$与具有如下单位脉冲响应

$$h_k(n)=W_N^{-kn}u(n) \tag{10.43}$$

的滤波器的卷积，该滤波器在 $n=N$ 点的输出就是 DFT 在频点 $\omega_k=\frac{2\pi}{N}k$ 的值，即

$$X(k)=y_k(n)|_{n=N} \tag{10.44}$$

我们可通过比较(10.41)和(10.42)来验证上式。

单位脉冲响应为 $h_k(n)$ 的这一滤波器的系统函数为

$$H_k(z)=\frac{1}{1-W_N^{-k}z^{-1}} \tag{10.45}$$

这个滤波器只有一个位于单位圆上频率为 $\omega_k=2\pi k/N$ 处的极点。因此,可使输入数据块通过 N 个并行的单极点滤波器(谐振器)组来计算全部 DFT,其中每个滤波器有一个位于 DFT 相应频率的极点。

我们可用(10.45)给出的滤波器差分方程以迭代方法计算 $y_k(n)$,来代替(10.42)那样卷积计算 DFT,于是我们有

$$y_k(n)=W_N^{-k}y_k(n-1)+x(n),\ y_k(-1)=0 \tag{10.46}$$

预期的输出为 $X(k)=y_k(N)$,为了执行该计算,可以只算一次相位因子 W_N^{-k},将其存贮起来。

将具有一对复共轭极点的谐振器组合在一起,就可避免(10.46)中的复数乘法和复数加法运算,这样就导出具有如下形式系统函数的双极点滤波器:

$$H_k(z)=\frac{1-W_N^kz^{-1}}{1-2\cos(2\pi k/N)z^{-1}+z^{-2}} \tag{10.47}$$

图 10.15 表示的该系统的实现的差分方程为:

$$v_k(n)=2\cos\frac{2\pi k}{N}v_k(n-1)-v_k(n-2)+x(n) \tag{10.48}$$

$$y_k(n)=v_k(n)-W_N^kv_k(n-1) \tag{10.49}$$

初始条件为 $v_k(-1)=v_k(-2)=0$。这就是 Goertzel 算法。

公式(10.48)中的递推关系对 $n=0,1,\cdots\cdots,N$ 重复 $N+1$ 次,但方程(10.49)仅在 $n=N$ 时刻计算一次,每次计算只需要计算一次实数乘和两次实数加。所以,对实数序列 $x(n)$,由于对称性,用这种算法求出 $X(k)$和 $X(N-k)$的值需要 $N+1$ 次实数乘法运算。

我们现在能够用 Goertzel 算法实现 DTMF 解码器了。由于有 8 种可能的音频要检测,所以需要 8 个(10.47)所给出的滤波器,每个滤波器调谐到这 8 个频率之一上。在 DTMF 解码器中,不需要计算复数值 $X(k)$,只要求出幅度值 $|X(k)|$ 或幅度平方值 $|X(k)|^2$ 就够了。所以,计算涉及分子项(滤波器计算的前向部分)的 DFT 值的最后一步可以简化,我们有

$$|X(k)|^2=|y_k(N)|^2=|v_k(N)-W_N^kv_k(N-1)|^2 \tag{10.50}$$

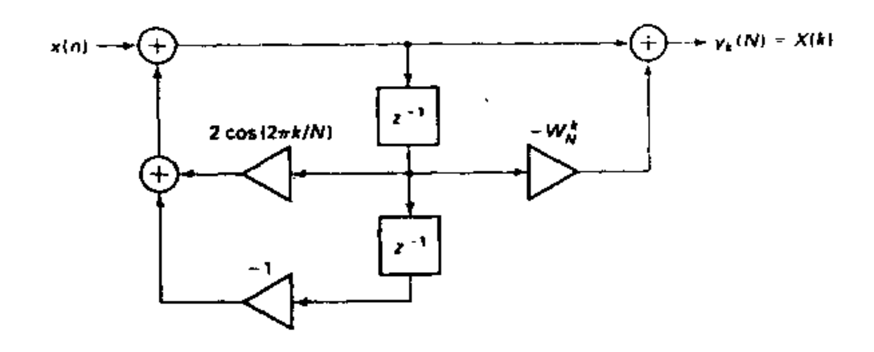


图 10.15 计算 DFT 的双极点谐振器实现

$$= v_k^2(N) + v_k^2(N-1) - (2\cos\frac{2\pi k}{N})v_k(N)v_k(N-1)$$

于是 DTMF 解码器中就完全避免了复数运算。

设计题 10.6 DTMF 拨号

本题的目的是理解 DTMF 音频产生软件和 DTMF 解码算法(Goertzel 算法),设计如下 MATLAB 程序模块。

1.音频产生函数:它接收电话拨号数列,产生包含适当音频(见图 10.14)的信号,每个数据信号持续半秒,采样频率为 8kHz。

2.拨号音频产生器:它在规定的持续时间内以 8kHz 的采样频率,产生(350+440)Hz 频率的样本。

3.实现(10.50)的解码函数:它接收 DTMF 信号,产生包含拨号数字的数列。

要求产生包含数字和拨号音频的几个拨号表矩阵,做音频产生和检测程序实验,并评论所观察的结果。用 MATLAB 的声音生成功能监听产生的音频,并观测所产生音频的成分。

二进制数据通信

用 PCM、ADPCM、DM 和 LPC 编码的数字化语音信号通常用数字调制方式传送到解码器。二进制数字通信系统用两种信号波形(如 $s_1(t) = s(t)$ 及 $s_2(t) = -s(t)$),传送代表语音信号的二进制序列。如果数据位为 1,则将信号波形 $s(t)$ (它在 $0 \leqslant t \leqslant T$ 区间不为零)传送到接收机,如果数据位为 0,则传送信号波形 $-s(t)$ ($0 \leqslant t \leqslant T$),时间间隔 T 称为信号间隔,而信道上的位率为 $R = 1/T$ 位/秒。典型的信号波形 $s(t)$ 为矩形脉冲,即 $s(t) = A$, $0 \leqslant t \leqslant T$,其能量为 $A^2 T$。

实际上,由于加性噪声和其他类型的信道失真使信道上传送的波形受到干扰,就是这些干扰最终限制了通信系统的性能,我们一般用平均误差概率(常称为位误差率)来度量通信系统性能。

设计题 10.7 二进制通信系统

本题的目的是用计算机仿真的方法研究加性噪声信道中二进制数据通信系统的性能,要仿真的基本系统结构如图 10.16,这需要五个 MATLAB 函数。

1.二进制数据产生器模块:它产生独立的等概率二进制序列。

2.调制器函数:将二进制 1 变换成 M 个连续 +1 的序列,将二进制 0 变换成 M 个连续 -1。这样 M 个 +1 表示一个采了样的矩形脉冲。

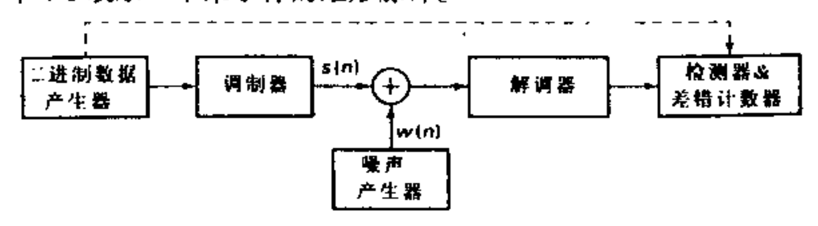


图 10.16 二进制数据通信系统模型

3.噪声产生函数:它产生在$(-a,a)$上均匀分布的数字序列,每个噪声样本要加到相应的信号样本上去。

4.解调器函数:它把从信道接收到的 M 个连续的已受干扰的 +1 或 -1 序列加起来。我们假设解调器是定时同步的,所以知道每个二进制波形的起点和终点。

5.检测器和差错计数模块:检测器将解调器的输出与零比较,如果输出大零则判定为 1,如果输出小于零,则判为 0。如果检测器输出与发送器传送的位不一致,则计数器计一个差错,差错率取决于 M 值与加性噪声功率(它是 $P_n=a^2/3$)之比(称之为信噪比)。

绘制所测量的差错率与不同的信噪比(可改变 M 保持P_n固定,或反之)的关系曲线。

扩展频谱通信

扩展频谱信号常用于因干扰而恶化的通信信道上来传送数字数据,信道干扰有人为干扰或来自信道其他用户的干扰(如蜂窝式电话或其他无线用户)。在通信以外的应用中,扩展频谱信号用于获得雷达和导航中距离(时延)和距离变化率(速度)的精确度量。限于篇幅,我们只讨论扩展频谱在数字通信中的应用,这种信号的特性是其频带宽度比信息速率(位/秒)大得多。

在对抗人为干扰时,企图中断通信的干扰者没有通信信号特性的先验知识,这对通信者而言是重要的。为了实现这一点,发信者在每个可能发送的信号波形中引入不可预测性或伪随机成分,这些引入的成分对通信系统指定的接收机是已知的,但干扰台不知道。因此,干扰台必须在不知道预期信号中的伪随机特性的条件下发送干扰信号。

在多路通信系统中,多个用户公用一个通信通道,因此出现了来自其他用户的干扰。在任何给定时间,这些用户中的一个子集可以通过公共信道同时发送信息到相应的接收机。为了使信道中传送的信号能相互区别,将不同的伪随机模板(称为多路码)叠加在每个发送信号中,所以只要知道伪随机模板,特定的接收机就能够恢复专门发给它的数据。这种通信技术(允许多个用户同时使用一个公用信道传送数据的技术)称为码分多址(CDMA)。

图 10.17 所示的方框图说明了扩频数字通信系统的基本组成,它与普通数字通信系统的差别在于包括两个相同的伪随机模板发生器,一个与发送端的调制器相接,第二个与接收端的解调器相接。该发生器产生伪随机或伪噪声(PN)二进制值序列(一串 ±1),它在调制器中被加到发送信号中,而在解调器中则被从接收信号中移去。

为了解调接收的信号,解调器中产生的 PN 序列必须与接收到的信号中包含的 PN 序列

同步。在发送数据之前先向接收机发一个短的固定的 PN 序列来建立同步，同步建立后，开始发送数据。

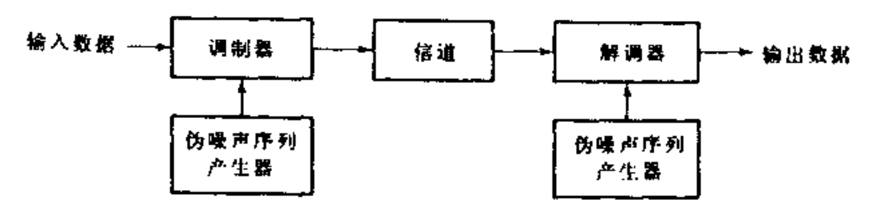


图 10.17 基本扩展频谱数字通信系统

设计题 10.8 二进制扩展频谱通信

本课题的目的是证明被正弦干扰淹没的 PN 扩展频谱信号的有效性。让我们考虑设计题 10.7 中描述的二进制通信系统。用二进制(± 1)PN 序列乘其调制器的输出，再用相同的二进制 PN 序列乘到达解调器的输入，这样就去掉了 PN 序列对预期信号的影响，信道对发送信号的干扰有加性宽带噪声序列$\{w(n)\}$和形为 $i(n)=A\sin\omega_0 n$(其中 $0<\omega_0<\pi$)的正弦干扰序列。假定 $A\geqslant M$，M 为对来自调制器中的每位的采样数。基本的二进制扩展频谱系统如图 10.18 所示。显然，这正是图 10.16 所示的二进制数字通信系统中加入了正弦干扰和 PN 序列产生器。可用一随机数发生器来产生等概率 ± 1 序列来构成 PN 序列.

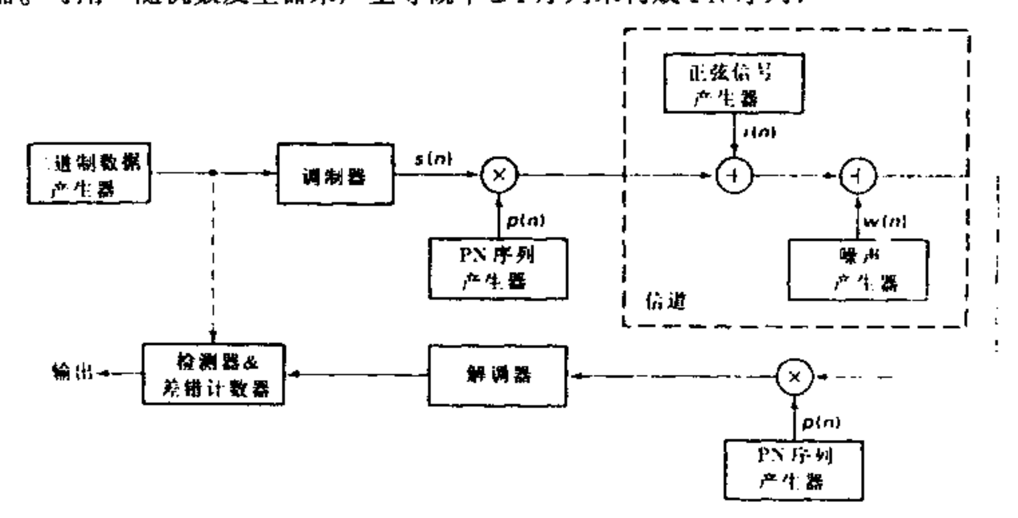


图 10.18 二进制 PN 扩展频谱系统仿真实验框图

在用 PN 序列和不用 PN 序列两种情况下运行仿真系统，并在 $A\geqslant M$ 的条件下对不同的 M 值(如 $M=50,100,500,1000$)测量差错率，解释 PN 序列对正弦干扰信号的影响，这说明了为什么扩展频谱系统在抗正弦干扰信号方面优于普通二进制通信系统。

小　　结

在本章中，我们集中讨论了波形表示与编码的应用，特别是介绍了几种把模拟波形数字化的方法，包括 PCM、DPCM、ADPCM、DM、ADM 和 LPC。这些方法已广泛应用于语音编码和传输中，包含这些波形编码方法的设计题都已用公式表示以便用 MATLAB 程序仿真。

我们还介绍了信号检测和通信系统，其中可以用 MATLAB 执行信号处理任务，也留下了有关这些应用的设计题。

附录 A PWS _ DSPC 汉化的数字信号处理工具箱 1.0 版中的 Matlab 函数

```
% 用于<数字信号处理及其 MATLAB 实现>一书
%
% 离散信号和系统
%      conv _ m       - 改进的线性卷积子程序 (P22)
%      conv _ tp      - 用 Toeplitz 矩阵计算的线性卷积(P34)
%      evenodd        - 将实信号分解为偶和奇两部分(P15)
%      impseq         - 产生脉冲序列:x(n) = delta(n - n0); n1 <= n,n0 <= n2
                        (P6)
%      sigadd         - 信号相加运算(P8)
%      sigfold        - 信号折叠运算(P10)
%      sigmult        - 信号乘法运算(P9)
%      sigshift       - 信号时移运算(P9)
%      stepseq        - 产生阶跃序列: x(n) = u(n - n0); n1 <= n,n0 <= n2(P6)
% 离散时间付利叶变换( z 变换)
%      pfe2rfz        - 在 z 域由部分分式展开为有理函数(ch.4)
%      rf2pfez        - 在 z 域由有理函数展开为部分分式(ch.4)
% 离散付利叶变换
%      circevod       - 实信号分解为循环偶分量和循环奇分量两部分(P132)
%      circonvt       - 时域中的循环卷积(P139)
%      cirshftt       - 时域中的循环移位(P135)
%      dfs            - 计算离散付利叶系数(P109)
%      dft            - 计算离散付利叶变换(P120)
%      hsolpsav       - 采用 FFT 高速分段卷积的重叠保留法(P157)
%      idfs           - 计算逆离散付利叶级数(P110)
%      idft           - 计算逆离散付利叶变换(P121)
%      mod            - 计算 m = n mod N = ((n)) (P119)
%      ovrlpsav       - 分段卷积的重叠保留法 (P147)
% 数字滤波器结构
%      cas2dir        - 级联到直接的形式转换(P173)
%      casfiltr       - IIR 和 FIR 滤波器的级联实现(P172)
%      cplxcomp       - 比较两个复数对(P176)
%      dir2cas        - 直接到级联的型式转换(P171)
%      dir2fs         - 直接形式到频率采样型的转换(P187)
```

```
%     dir2ladr      - IIR 直接形式极__零点到格型/梯形的转换(P199)
%     dir2latc      - FIR 直接形式到全零点格型形式的转换(P193)
%     dir2par       - 直接形式到并联形式的转换(P175)
%     dir2par0      - 用旧版信号处理工具箱时的 dir2par 函数
%     ladr2dir      - 格型/梯形形式到 IIR 直接形式的转换(P199)
%     ladrfilt      - 格型/梯形形式的 IIR 滤波器实现(P200)
%     latc2dir      - 全零点格型形式到 FIR 直接形式的转换(P194)
%     latcfilt      - FIR 滤波器的格型形式的实现(P194)
%     par2dir       - 并联形式到直接形式的转换(P177)
%     parfiltr      - IIR 滤波器的并联形式的实现(P177)
% FIR 滤波器设计
%     amp_res       - 由 FIR 滤波器脉冲响应求其幅频特性(P271)
%     blackman      - 布莱克曼窗函数(P230)
%     freqz_m       - 改进型的 freqz 子程序(P233)
%     Hr_Type1      - 计算 1 型 FIR 低通滤波器(P215)
%     Hr_Type2      - 计算 2 型 FIR 低通滤波器(P215)
%     Hr_Type3      - 计算 3 型 FIR 低通滤波器(P216)
%     Hr_Type4      - 计算 4 型 FIR 低通滤波器(P216)
%     ideal_lp      - 理想低通滤波器脉冲响应计算 (P232)
% IIR 滤波器设计
%     afd_butt      - 模拟低通巴特沃思滤波器设计(P286)
%     afd_chb1      - 模拟低通切比雪夫Ⅰ型滤波器设计(P292)
%     afd_chb2      - 模拟低通切比雪夫Ⅱ型滤波器设计(P295)
%     afd_elip      - 模拟椭圆低通滤波器设计(P299)
%     cheb1hpf      - 用切比雪夫Ⅰ型原型作 IIR 高通滤波器设计(P330)
%     freqs_m       - 改进型的 freqs 子程序(P286)
%     imp_invr      - 由模拟到数字滤波器的脉冲响应不变变换(P303)
%     sdir2cas      - s 平面的直接形式到级联形式的变换(P282)
%     u_buttap      - 未归一化的巴特沃思模拟低通滤波器原型(P282)
%     u_chb1ap      - 未归一化的切比雪夫Ⅰ型模拟低通滤波器原型(P290)
%     u_chb2ap      - 未归一化的切比雪夫Ⅱ型模拟低通滤波器原型(P294)
%     u_elipap      - 未归一化的椭圆模拟低通滤波器原型(P298)
%     zmapping      - z 域中的频带变换(P326)
% 自适应滤波
%     lms           - 系数调整的 LMS 算法(P347)
% 数字通信
%     mulaw_c       - mu 规则压缩器(式(10.5))
%     mulaw_e       - mu 规则扩张器(式(10.7))
%     quantize      - 将信号量化为 B 位(图 10.2)
```

```
% 杂项
%     contents      - 内容文件(你正在读的)
%     db2delta      - 由相对的 dB 数转换为绝对的 delta 数.
%     delta2db      - 由绝对的 delta 数转换为相对的 dB 数
%     pzplotz       - 按正方座标画出 z 平面上单位圆及零极点分布图
%     sinc          - sinc(x) = sin(pi * x)/(Pi * x)
```

注:各行后面括号内为源程序所在页码,无页码者说明书中无源程序,需从本书所配软盘中寻找。

参考文献

[1] *MATLAB Reference Guide: High-Performance Numeric computation and Visualization Software*. The MathWorks. Inc., South Natick. MA. 1984-1994.

[2] *MATLAB User's Guide: High performance Numeric Computation and Visualization Software*. The MathWorks. Inc., South Natick, MA. 1984 – 1994.

[3] *The Math Works, Inc: The Student Edition of MATLAB*. Prentice Hall. Englewood Cliffs, NJ, version 4 edition, 1995.

[4] J. W. Cooley and J. W. Tukey. An algorithm for the machine computation of complex Fourier series. *Mathematical Computations*. 19:297 301, April 1965.

[5] C. de Boor. *A Practical Guide to Splines*. Springer-Verlag, 1978.

[6] J. L. Flanagan et al. Speech coding. *IEEE Transactions on Communications*, COM-27: 710-736, April 1979.

[7] D. A. George, R. R. Bowen, and J. R. Storey. An adaptive decsion feedback equalizer. *IEEE Transactions on Communications Technology*, pages 281-293, June 1971.

[8] J. A. Greefkes. A digitally companded delta modulation modem for speech transmission. In *Proceedings of IEEE International Conference on Communications*, pages 7.33-7.48, June 1970.

[9] S. Haykin. *Adqptive Filter Theory*. Prentice Hall, Englewood Cliffs, NJ, 1986.

[10] F. M. Hsu and A. A. Giordano. Digital whitening techniques for improving spread spectrum communications performance in the presence of narrowband jamming and interference. *IEEE Transactions on Communications*, COM-26: 209-216, February 1978.

[11] V. K. Ingle and J. G. Proakis. *Digital Signal Processing using the ADSP*-2101. Prentice Hall, Englewood Cliffs, NJ. 1991.

[12] N. S. Jayant. Adaptive delta modulation with one-bit memory. *Bell System Technical Journal*, pages 321-342, March 1970.

[13] N. S. Jayant. Digital coding of speech waveforms: Pcm, dpcm and dm quantizers. *proceeding of the IEEE*, 62:611-632, May 1974.

[14] J. W. Ketchum and J. G. Proakis. Adaptive algorithms for estimation and suppression of narrowband interference in pn spread-spectrum systems. *IEEE Transactions on Communications*, COM-30:913-922, May 1982.

[15] N. Levinson. The Wiener rms(root-mean-square)error criterion in filter design and prediction, *Journal of Mathematical Physics*, 25:261-278, 1974.

[16] A. V. Oppenheim and R. W. Schafer. *Discrete-Time Signal Processing*. Prentice Hall, Englewood Cliffs, New Jersey, 1989.

[17] T. W. Parks and J. H. McClellan. A program for the design of linear-phase finite impulse response digital filters. *IEEE Transactions on Audio and Electroacoustics*, AU-20:195-199, August 1972.

[18] J. G. Proakis. *Digital Communications*. McGraw-Hill, New York, NY, third edition, 1995.

[19] J. G. Proakis and D. G. Manolakis. *Digital Signal Processing: Principles, Algorithms and Applications*. Macmillan, New York, NY, third edition, 1996.

[20] L. R. Rabiner and B. Gold. *Theory and Applications in Digital Signal Processing*. Prentice Hall, Englewood Cliffs, NJ, 1975.

[21] L. R. Rabiner, R. W. Schafer, and C. A. McGonegal. An approach to the approximation problem for nonrecursive digital filters. *IEEE Transactions on Audio and Electroacoustics*, AU-18:83-106, June 1970.

[22] B. Widrow et al. Adaptive noise cancelling: Principles and applications. *Proceedings of the IEEE*, 63:1692-1716, December 1975.

[23] B. Widrow, P. Manley, and L. J. Griffiths. Adaptive antenna systems. *Proceedings of the IEEE*, 55:2143-2159, December 1967.

